



UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

FACULTAD DE INGENIERIA

SISTEMA EXPERTO PARA MUESTREO DE ACEPTACION
EN PROGRAMAS DE CONTROL DE CALIDAD

T E S I S
PARA OBTENER EL TITULO DE:
INGENIERO EN COMPUTACION
P R E S E N T A
EVA GABRIELA ELIZALDE GUALITO

DIRECTOR DE TESIS: DR. ARTURO GUILLERMO REINKING CEJUDO



MEXICO, D. F.

OCTUBRE, 2004



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.

INDICE

Introducción.....	1
Capítulo 1: Sistemas Expertos.....	3
1.1 Desarrollo de los sistemas expertos.....	3
1.2 Función de los sistemas expertos.....	4
1.3 Componentes de un sistema experto.....	5
1.3.1 Base de conocimiento.....	6
1.3.1.1 Base de hechos.....	8
1.3.2 Motor de inferencia.....	8
1.3.2.1 Encadenamiento hacia delante.....	9
1.3.2.2 Encadenamiento hacia atrás.....	9
1.3.2.3 Reglas de producción.....	9
1.3.3 Componente explicativo.....	9
1.3.4 Interfase de usuario.....	10
1.3.5 Componente de adquisición.....	12
1.4 Herramientas para el desarrollo de sistemas expertos.....	13
1.5 El equipo de desarrollo.....	14
1.5.1 El experto.....	15
1.5.2 El ingeniero de conocimiento.....	15
1.5.3 El usuario.....	15
1.6 Ventajas de los sistemas expertos.....	15
1.7 Campos de aplicación.....	16
Capítulo 2: Control Estadístico de Calidad.....	19
2.1 Control estadístico del proceso.....	19
2.2 Muestreo para aceptación.....	19
2.2.1 La curva OC.....	20
2.2.2 Planes de muestreo de atributos.....	24
2.2.2.1 Variables y atributos.....	25
2.2.2.2 Tipos de muestreo en los planes de muestreo por atributos.....	25
2.2.3 Datos históricos de los sistemas de muestreo para aceptación basados en el concepto de AQL.....	27
2.2.4 Descripción de la norma militar estándar (MIL-STD-105d).....	27
2.2.4.1 Uso de la norma militar estándar	29
2.2.4.2 Texto de la norma militar estándar	29
2.2.5 Criterios para el cambio de inspección normal, rigurosa y reducida.....	31
2.2.5.1 Reglas de cambio de inspección normal, rigurosa y reducida en el shell i2+.....	33

Capítulo 3: Selección de los Recursos.....	39
3.1 Necesidades.....	39
3.2 Análisis de recursos.....	39
3.2.1 Bases de datos.....	39
3.2.1.1 Ejemplos de bases de datos.....	40
3.2.2 Lenguajes de programación.....	41
3.2.2.1 Ejemplos de lenguajes de programación.....	41
3.2.3 Sistemas operativos.....	43
3.2.3.1 Ejemplos de sistemas operativos.....	44
3.2.4 Shell para sistemas expertos.....	45
3.2.4.1 Ejemplos de shell.....	45
3.3 Selección de los recursos a utilizar.....	46
3.3.1 Windows 98 como sistema operativo.....	46
3.3.2 Visual Basic 6.0 como entorno de desarrollo.....	47
3.3.3 ACCESS 2000 Y EXCEL 2000 como bases de datos.....	48
3.3.4 Clips como máquina de inferencias.....	48
3.3.4.1 Clips activeX ocx	49
Capítulo 4: Entorno de Desarrollo del Sistema Experto.....	51
4.1 Elementos de programación de clips.....	51
4.2 Comunicación interna del shell clips.....	53
4.3 Base de conocimiento del sistema experto muestreo para aceptación.....	54
4.4 Relación entre Clips, activeX ocx Y Visual Basic.....	61
4.5 Comunicación externa del shell Clips.....	64
4.6 Análisis del sistema.....	65
4.6.1 DFD del sistema de nivel 0.....	65
4.6.2 DFD del sistema de nivel 1.....	66
4.6.3 DFD del sistema de nivel 2 para el proceso “registrar un producto nuevo”..	67
4.6.4 DFD del sistema de nivel 2 para el proceso “ acceder a un producto existente”.....	69
4.6.5 DFD del sistema de nivel 2 para el proceso “descontinuar un producto”	71
Capítulo 5: Ejemplos de la Aplicación del Sistema Experto.....	73
5.1 Pantalla principal.....	73
5.2 Ejemplo de la aplicación del sistema experto para dar de alta un producto nuevo.....	83
5.3 Ejemplo de la aplicación del sistema experto en el cambio de inspección normal a reducida en un producto existente.....	91
5.4 Ejemplo de la aplicación del sistema experto en el cambio de inspección reducida a normal.....	97
5.5 Comentarios finales.....	100
Capítulo 6: Mantenimiento.....	101

Capítulo 7: Conclusiones.....102
Bibliografía.....104
Anexos.....105

DEDICATORIAS

*A Dios por darme la oportunidad
de alcanzar este fin.*

*A mis padres y principalmente a mi madre, Gracias
Por haberme brindado su apoyo económico, moral,
e incondicional a lo largo de mi vida.*

*A mi tía Marcela por brindarme su amistad
Y su cariño así como a mis hermanos y amigos.
A Ramiro Por su ejemplo de constancia y dedicación.*

*A mis maestros por orientarme Y
prepararme para mi vida laboral
Y profesional.*

*A la Universidad Nacional Autónoma De México
Por haberme recibido en sus aulas.*

*A mi asesor de tesis Dr. Arturo Reinking por
Todo lo que me aportó en la realización de este trabajo.*

INTRODUCCIÓN

En situaciones recurrentes como es el caso de la aprobación de lotes de elementos o productos con estándares de calidad en la forma de pasa no pasa, se tiene que manejar gran cantidad de información, lo anterior motiva un proyecto que facilite el manejo de muchos datos y de procesos repetitivos, y que dé lugar a un software amigable y confiable que trabaje por nosotros proporcionándole solo los datos estrictamente necesarios.

Esta tesis es un ejemplo práctico que muestra como un sistema experto puede de manera sencilla tomar el conocimiento teórico de un grupo de reglas de cambio de inspección necesarias en la aprobación de lotes con estándares de calidad. Se muestra que es fácil pasar el conocimiento heurístico a una base de conocimiento de algún sistema experto, que logra dar los mismos resultados de este cambio de inspección, ya sea de inspección normal, inspección reducida o inspección rigurosa sin caer en errores por parte del usuario en la aplicación del estándar y en el seguimiento de los procedimientos.

Además se incorporan las bases de datos con toda la información del estándar estudiado (MIL-STD-105D) que frecuentemente resulta tedioso y da lugar a errores por parte del usuario. Con este sistema los errores de la administración de ésta gran cantidad de información y de datos son eliminados.

En esta tesis se estudia en el capítulo uno que son los sistemas experto, como trabajan, en que campos pueden desarrollarse, sus ventajas y sus desventajas dando un resumen adecuado y completo.

Por otra parte, para entender el problema al cual se quiere dar solución a través de un sistema experto, en el capítulo dos se analiza el control estadístico de calidad refiriéndose principalmente al problema de muestreo para aceptación tomando al estándar MIL-STD-105D para su resolución y llevándolo a la practica con el software desarrollado.

Para éste desarrollo se cuenta en principio con diferentes tipos de paquetería y de entre las cuales se tuvo que hacer una selección. En el capítulo tres se describe algunas de sus características, fundamentando las decisiones para descartar unas y seleccionar otras.

Debido que la base de conocimiento del sistema se programo en Clips en el capítulo cuatro se estudia el shell Clips y sus elementos de programación. Se incluye además la base de conocimiento del sistema experto muestreo para aceptación y se describe Activex OCX, herramienta necesaria para manejar los datos de la base de conocimiento de Clips y poder visualizarlos en Visual Basic en pantallas de fácil acceso y características agradables a la vista. Se muestran también los diagramas de flujo de datos del sistema desarrollado, que nos permiten ver al sistema como una red de procesos funcionales.

En el capítulo cinco se ejemplifica el proceso de utilización del programa desarrollado, mostrando las diferentes pantallas con las que el usuario deberá interactuar al aplicar el sistema. Los ejemplos abarcan diferentes etapas de inicio y su interfase muestra las soluciones que se obtuvieron al registrar los diferentes datos proporcionados por el usuario.

Por último el capítulo seis nos da un perspectiva de los logros y las conclusiones obtenidas que resultaron del desarrollo del sistema experto “muestreo para aceptación en programas de control de calidad”.

CAPITULO 1

SISTEMAS EXPERTOS

Bajo el término de Sistemas Expertos se entiende una clase de aplicaciones de la Inteligencia Artificial que imitan el comportamiento de un experto humano en la solución de problemas recurrentes. Pueden almacenar conocimientos de expertos para un campo determinado y solucionar un problema mediante el encadenamiento de deducciones lógicas.

Algunos especialistas opinan que los sistemas expertos se pueden considerar como el primer producto verdaderamente operacional de la Inteligencia Artificial. Son programas de computadora diseñados para actuar como un especialista humano en un dominio particular o área de conocimiento. En este sentido, pueden considerarse como intermediarios entre el experto humano, que transmite su conocimiento al sistema, y el usuario que lo utiliza para resolver un problema con la eficacia del especialista. El sistema experto utilizará para ello el conocimiento que tenga almacenado y algunos métodos de inferencia.

A la vez, el usuario puede aprender observando el comportamiento del sistema. Es decir, los sistemas expertos se pueden considerar simultáneamente como un medio de ejecución y transmisión del conocimiento.

Lo que se intenta, de esta manera, es representar los mecanismos heurísticos que intervienen en un proceso de solución de problemas. Éstos mecanismos forman ese conocimiento difícil de expresar que permite que los expertos humanos sean eficaces calculando lo menos posible. Los sistemas expertos contienen ese "saber hacer".

La característica fundamental de un sistema experto es que separa los conocimientos almacenados (base de conocimiento) del programa que los controla (motor de inferencia). Los datos propios de un determinado problema se almacenan en una base de datos aparte (base de hechos).

Una característica adicional deseable, y a veces fundamental, es que el sistema sea capaz de justificar su propia línea de razonamiento de forma inteligible por el usuario.

1.1 DESARROLLO DE LOS SISTEMAS EXPERTOS.

Los sistemas expertos proceden inicialmente de la inteligencia artificial a mediados de los años sesenta. En ese período se creía que bastaban unas pocas leyes de razonamiento junto con potentes computadoras para producir resultados brillantes. Un intento en ese sentido fue el llevado a cabo por los investigadores Alan Newell y Herbert Simon que desarrollaron un programa denominado GPS (General Problem Solver; solucionador general de problemas). Podía trabajar con criptoaritmética, con las torres de Hanoi y con otros problemas similares. Lo que no podía hacer el GPS era resolver problemas del mundo real, tales como un diagnóstico médico.

Algunos investigadores decidieron entonces cambiar por completo el enfoque del problema restringiendo su ambición a un dominio específico e intentando simular el

razonamiento de un experto humano. En vez de dedicarse a computarizar la inteligencia general, se centraron en dominios de conocimiento muy concretos. De esta manera nacieron los sistemas expertos.

A partir de 1965, un equipo dirigido por Edward Feigenbaum, comenzó a desarrollar sistemas expertos utilizando bases de conocimiento definidas minuciosamente.

En 1967 se construye DENDRAL, que se considera como el primer sistema experto. Se utilizaba para identificar estructuras químicas moleculares a partir de su análisis espectrográficos.

Entre 1970 y 1980 se desarrolló MYCIN para consulta y diagnóstico de infecciones de la sangre. Este sistema introdujo nuevas características: utilización de conocimiento impreciso para razonar y posibilidad de explicar el proceso de razonamiento. Lo más importante es que funcionaba de manera correcta, dando conclusiones análogas a las que un ser humano daría tras largos años de experiencia. En MYCIN aparecen claramente diferenciados el motor de inferencia y la base de conocimientos. Al separar esas dos partes, se puede considerar el motor de inferencias aisladamente. Esto da como resultado un sistema vacío o shell (cascarón). Así surgió EMYCIN (MYCIN Esencial) con el que se construyó SACON, utilizado para estructuras de ingeniería, PUFF para estudiar la función pulmonar y GUIDON para elegir tratamientos terapéuticos.

En esa época se desarrollaron también: HERSAY, que intentaba identificar la palabra hablada, y PROSPECTOR, utilizado para hallar yacimientos de minerales. De este último se derivó el shell KAS (Knowledge Acquisition System).

A partir de 1980 se ponen de moda los sistemas expertos y numerosas empresas de alta tecnología investigan en ésta área de la inteligencia artificial, desarrollando sistemas expertos para su comercialización. Se llega a la conclusión de que el éxito de un sistema experto depende casi exclusivamente de la calidad de su base de conocimiento. El inconveniente es que codificar la pericia de un experto humano puede resultar difícil, largo y laborioso.

Un ejemplo de sistema experto moderno es CASHVALUE, que evalúa proyectos de inversión y VATIA, que asesora acerca del impuesto sobre el valor agregado o I.V.A.

1.2 FUNCIÓN DE LOS SISTEMAS EXPERTOS.

Los Sistemas Expertos es uno de los temas que componen las investigaciones en el campo de la Inteligencia Artificial. Un sistema de computadoras que trabaje con técnicas de Inteligencia Artificial deberá estar en situación de combinar información de forma “inteligente”, alcanzar conclusiones y justificarlas (al igual que el resultado final). Los Sistemas Expertos son una expresión de los sistemas basados en el conocimiento. Con la aplicación de técnicas de Inteligencia Artificial se inicia la transición del procesamiento de datos al procesamiento de conocimientos.

Los sistemas expertos se aplican por norma general en problemas que implican un procedimiento basado en el conocimiento. Un procedimiento de solución basado en el conocimiento comprende las siguientes capacidades:

- Utilización de normas o estructuras que contengan conocimientos y experiencias de expertos especializados.
- Deducción lógica de conclusiones.
- Capaz de interpretar datos ambiguos.
- Manipulación de conocimientos afectados por valores de probabilidad.

La función de un Sistema Experto es la de aportar soluciones a problemas, como si de humanos se tratara, es decir capaz de mostrar soluciones inteligentes. Esto es posible gracias a que al sistema lo crean con expertos (humanos), que intentan estructurar y formalizar conocimientos poniéndolos a disposición del sistema, para que este pueda resolver una función dentro del ámbito del problema, de igual forma que lo hubiera hecho un experto.

Acceder a los conocimientos adquiridos por experiencia es lo más difícil, ya que los expertos, al igual que otras personas, apenas los reconocen como tales. Son buscados con mucho esfuerzo y cuidado siendo descubiertos de uno en uno, poco a poco.

1.3 COMPONENTES DE UN SISTEMA EXPERTO.

Sus dos componentes principales son la base de conocimientos y un motor de inferencias. El conocimiento se representa de diversas maneras, que se enumeran en la sección 1.3.5, para crear bases de conocimiento.

Una vez creada la base de conocimiento, se debe diseñar un método para utilizarla, que es el motor de inferencia. El motor de inferencia manipula la información simbólica almacenada en la base de conocimiento mediante un proceso de búsqueda como se verá mas adelante.

La base de conocimiento es una base de datos que posee una información y unas reglas específicas sobre una materia determinada.

Objeto: La conclusión que es definida por sus reglas asociadas.

Atributo: Una cualidad específica que, con su regla, ayuda a definir el objeto.

En base a objeto y atributo definimos a la base de conocimiento como una lista de objetos con sus reglas y atributos asociados.

Una característica decisiva de los Sistemas Expertos es la separación entre conocimiento (reglas, hechos) por un lado y su procesamiento por el otro. A ello se añade una interfase de usuario y un componente explicativo.

A continuación se da una breve descripción de cada uno de los componentes los cuales se muestran en la figura 1.3.1:

1- **La Base de Conocimientos** de un Sistema Experto contiene el conocimiento de los hechos y de las experiencias de los expertos en un dominio determinado.

2- **El Motor de Inferencia** de un Sistema Experto puede simular la estrategia de solución de un experto.

3- *El Componente Explicativo* explica al usuario la estrategia de solución encontrada y el porqué de las decisiones tomadas.

4- *La Interfase de Usuario* sirve para que éste pueda realizar una consulta en un lenguaje lo más natural posible.

5- *El Componente de Adquisición* ofrece ayuda a la estructuración e implementación del conocimiento en la base de conocimientos.

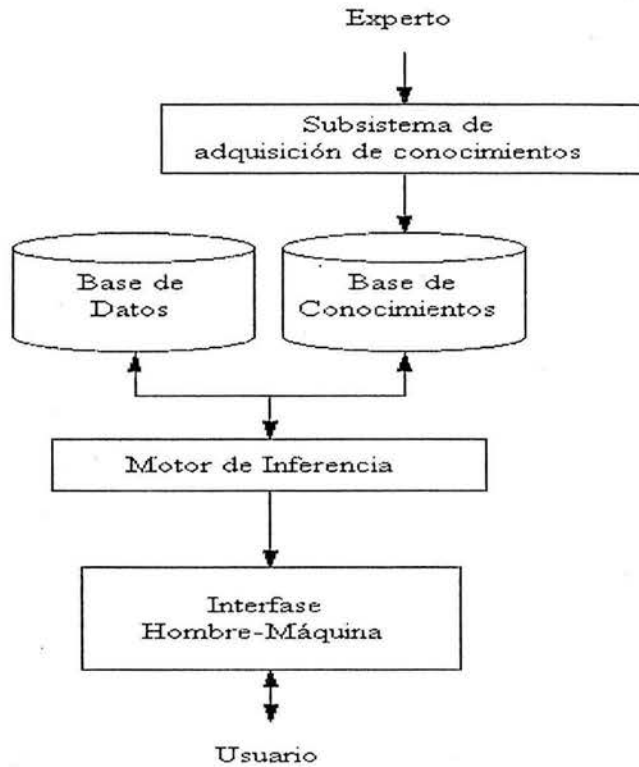


Figura1.3.1

A continuación se exponen mas en detalle los elementos que componen un sistema experto y que se muestra en la figura 1.3.1 anterior.

1.3.1 BASE DE CONOCIMIENTO.

Es donde se almacena el conocimiento en el campo a desarrollar. Los conocimientos que se suelen almacenar son de cuatro tipos: *Conocimiento objetivo*, que describe la situación real del sistema; *Conocimiento de sucesos*, relacionado con los sucesos que ocurren en el tiempo de ejecución; *Conocimiento del funcionamiento* del sistema, de cómo se hacen las cosas, y *Metaconocimiento*, que relaciona los anteriores.

La Base de conocimientos contiene todos los hechos, las reglas y los procedimientos del dominio de aplicación que son importantes para la solución del problema.

Para entender la base de conocimientos se dará el siguiente ejemplo: Los hechos son del tipo: **El velero "Tiburón"** tiene una longitud de 6m.

La representación de este conocimiento puede realizarse orientándola, por ejemplo, según objetos. Los objetos de una base de conocimientos pueden ser entonces: barco, barco a motor, barco a vela. Estos objetos están relacionados de tal forma que un barco a vela tiene todas las cualidades de un barco, y además todas las cualidades específicas de un barco a vela.

Todas las cualidades de un barco, por ejemplo: Desplazamiento sobre el agua, vienen descritas con el "barco". A través de la relación formulada, el barco a vela "hereda" estas cualidades, de forma que sólo hará falta describir sus cualidades particulares.

Este tipo de programación se define como programación orientada a objetos y se utiliza con frecuencia en el desarrollo de los Sistemas Expertos. Puede darse el caso de que determinados procesos y funciones deban subordinarse a unos objetos en particular, por ejemplo la velocidad como función de la fuerza y la dirección del viento. La velocidad se determinará en función de los datos particulares.

Cómo se lleva a cabo la clasificación en grupos de las características y de los procedimientos alrededor de un objeto con las técnicas de programación, y cómo deben ser las relaciones entre los objetos pueden variar mucho de aplicación a aplicación. Junto a estos objetos, la base de conocimientos dispone de reglas. Estas reglas se representan en forma de:

**Si premisas
Entonces Conclusión y/o Acción.**

En la zona de las premisas se solicitan vinculaciones lógicas referentes a las cualidades de los objetos.

En la zona de la conclusión se añaden nuevos hechos y cualidades a la base de conocimientos y/o se ejecutan acciones. Esto se define a menudo como programación orientada a reglas.

Como se puede deducir se plantean muchas preguntas para realizar la labor, como por ejemplo:

- ¿Qué objetos serán definidos ?
- ¿Cómo son las relaciones entre los objetos ?
- ¿Cómo se formularán y procesarán las reglas ?
- ¿La base de conocimientos hace totalmente referencia a la solución del problema?
- ¿La base de conocimientos es consistente?

Las respuestas a estas preguntas son el trabajo del Ingeniero del conocimiento junto con la colaboración de los expertos.

1.3.1.1 BASE DE HECHOS.

Representan el conocimiento del estado del sistema en un cierto instante. Usualmente se representa en una base de datos, y su información está directamente enlazada con la base de conocimientos. Un hecho puede provocar el disparo ("trigger") de una regla: (la base de hechos, almacena los hechos)

Hecho: Utilización_Máquina =0.873

Regla: SI Utilización_Máquina > 0.8

ENTONCES Estado_Máquina ="SOBRECARGADO".

Según esta regla, como Utilización_Máquina es mayor a 0.8, se cumple la premisa, se dispara la regla y se agrega un nuevo hecho Estado_Maquina = "SOBRECARGADO" a la base de hechos.

1.3.2 MOTOR DE INFERENCIA.

El motor de inferencias es la unidad lógica con la que se extraen conclusiones de la base de conocimientos, según un método fijo de solución de problemas que está configurado, imitando el procedimiento de los expertos humanos para solucionar problemas.

Una conclusión se produce mediante aplicación de las reglas sobre los hechos presentes.

Ejemplo:

Una Regla es : Si p y q entonces r

Se dan los hechos : p y q

p y q son justo aquellos hechos que se mencionan en la cláusula "si" de la regla, es decir, las condiciones para la aplicabilidad de la regla.

Aplicar la regla es: deducir de los hechos p y q implican el hecho r.

En un Sistema Experto existirá un hecho sólo cuando esté contenido en la base de conocimientos (en la base de hechos).

Los hechos que constan en la cláusula "si" se llaman *premisas*, y el contenido en la cláusula "entonces" se llama *conclusión*. Cuando se aplica una regla sobre algunos hechos cualesquiera se dice que se *dispara*. El disparo de una regla provoca la inserción del nuevo hecho en la base de conocimientos (en la base de hechos) como se explicó anteriormente.

Las funciones del motor de inferencia son:

- 1- Determinación de las acciones que tendrán lugar, el orden en que lo harán y cómo lo harán entre las diferentes partes del Sistema Experto.
- 2- Determinar cómo y cuándo se procesarán las reglas, y dado el caso también la elección de qué reglas deberán procesarse.
- 3- controlar el diálogo con el usuario.

La decisión sobre los mecanismos de procesamiento de reglas, es decir, qué estrategias de búsqueda se implementarán, es de vital importancia para la efectividad del sistema en su conjunto.

Ante problemas o clases de problemas distintos se estructuran, como es lógico, diferentes mecanismos de inferencia. El mecanismo de inferencia debe de estar "adaptado" al problema a solucionar. Una imposición de tipo monetario exige, bajo ciertas circunstancias, una estrategia distinta de procesamiento del conocimiento que un diagnóstico de causas de fallas de máquinas.

El motor de inferencia es la parte del sistema experto que intenta utilizar la información que el usuario le suministra para encontrar un objeto que le encaje. Hay dos categorías muy amplias sobre los motores de inferencia: la determinística y la probabilística. Para entender la diferencia entre estas categorías, imaginemos a dos especialistas uno en química y otro en socióloga.

El químico puede informar con certeza que si un átomo en cuestión tiene dos protones, entonces se trata de un átomo de helio. No hay problema acerca de la identidad del átomo porque el número de protones determina el tipo de elemento. Sin embargo, si se le pregunta a un sociólogo cuál es la mejor forma de evitar que los estudiantes abandonen la escuela, el sociólogo le dará una respuesta que califica como sólo probable, o como mucho, con una cierta probabilidad de éxito. Por lo tanto, la respuesta es probable, pero incierta.

Se pueden mencionar tres formas básicas de construir el motor de inferencia

1. Encadenamiento hacia adelante
2. Encadenamiento hacia atrás
3. Reglas de producción

1.3.2.1 ENCADENAMIENTO HACIA DELANTE.

Al encadenamiento hacia adelante se le llama algunas veces <<conducido por datos>> porque el motor de inferencia utiliza la información que el usuario le proporciona para moverse a través de un árbol de decisiones (ANDs) y (ORs) lógicos hasta que encuentra un punto terminal, que es el objeto. Si el motor de inferencia no puede encontrar un objeto usando la información existente, entonces le pide al usuario información adicional. Los atributos que definen al objeto crean un camino que conduce al mismo objeto: la única forma de alcanzar dicho objeto es la de satisfacer todas sus reglas. Por tanto, un motor de inferencia de encadenamiento hacia adelante comienza con alguna información y luego intenta encontrar un objeto que encaje con dicha información.

En resumen, en el *Encadenamiento hacia adelante*: se extraen conclusiones a partir del cumplimiento de las premisas de ciertas reglas que, a su vez, provocarán el cumplimiento de las premisas en otras reglas hasta ya no se puedan extraer mas conclusiones.

Definición: Llamada Forward Chaining, es un razonamiento dentro de un sistema lógico que avanza desde los datos o premisas hacia las conclusiones cuya validez aún no se sabe.

1.3.2.2 ENCADENAMIENTO HACIA ATRÁS.

El encadenamiento hacia atrás es el contrario al encadenamiento hacia adelante. Un motor de inferencia de encadenamiento hacia atrás comienza con una hipótesis (un objeto) y pide información para confirmarlo o negarlo. Al encadenamiento hacia atrás se le llama algunas veces <<conducido por objetos>> porque el sistema experto empieza con un objeto e intenta verificarlo.

Encadenamiento hacia atrás: se trata de validar las premisas de una regla y, como consecuencia, se van disparando aquellas reglas que validan las premisas de la regla original.

Definición: Llamado backchaining o Backward Chaining, esta forma de razonar dentro de un sistema lógico se basa en seguir la ruta desde lo que se quiere sustentar en su validez (un teorema) hacia los datos (las premisas). Aquí se parte de algo que se quiere demostrar y se buscan las oraciones que permitirían llegar a esa conclusión. Por último, se busca establecer las premisas correspondientes.

1.3.2.3 REGLAS DE PRODUCCIÓN.

Un motor de inferencia de reglas de producción es teóricamente superior tanto a un sistema de encadenamiento hacia adelante como hacia atrás porque pide la información que tenga una mayor importancia de acuerdo con el estado actual del sistema. Un motor de inferencia de reglas de producción es, en realidad, un motor de encadenamiento hacia atrás mejorado. La teoría operativa general es que el sistema pide como nueva información aquella que elimine la mayor incertidumbre posible del sistema.

Por ejemplo: para entender el funcionamiento de las reglas de producción, imagine que ha llamado a un médico porque su hijo está enfermo. El doctor, primero le pregunta si el niño tiene fiebre, porque la respuesta a esta pregunta reduce el mayor número de posibilidades. Si su respuesta es <<sí>>, el doctor le pregunta si su hijo tiene angustia. Al igual que en la pregunta anterior, el doctor se la hace antes que cualquier otra porque es la de mayor impacto en el diagnóstico, dado el estado actual. Este procedimiento continúa hasta que el doctor pueda dar un diagnóstico.

En este ejemplo, el punto clave estriba en que el doctor selecciona aquellas preguntas que le llevan más rápidamente hacia una conclusión.

1.3.3 COMPONENTE EXPLICATIVO.

Las soluciones descubiertas por los expertos deben poder ser repetibles tanto por el ingeniero del conocimiento en la fase de comprobación así como por el usuario. La exactitud de los resultados sólo podrá ser controlada, naturalmente, por los expertos.

Siempre es deseable que durante el trabajo de desarrollo del sistema se conozca el grado de progreso en el procesamiento del problema. Como se ha dicho en anterioridad pueden surgir unas preguntas como las siguientes:

- ¿Qué preguntas se plantean y por qué?
- ¿Cómo ha llegado el sistema a soluciones intermedias?
- ¿Qué atributos tienen los distintos objetos?

Los elementos del componente explicativo permiten seguir el progreso de la consulta al sistema de forma gráfica. Además los componentes explicativos intentan justificar su función rastreando hacia atrás el camino de la solución. Aunque encontrar la forma de representar finalmente en un texto lo suficientemente inteligible las relaciones encontradas depara las mayores dificultades. Los componentes explicativos pueden ser suficientes para el ingeniero del conocimiento, ya que está muy familiarizado con el entorno del procesamiento de datos, y a veces bastan también para el experto; pero para el usuario, que a menudo desconoce las sutilezas del procesamiento de datos, los componentes explicativos existentes son todavía poco satisfactorios.

1.3.4 INTERFASE DE USUARIO.

Su función es establece una comunicación sencilla entre el usuario y el sistema. El usuario puede consultar al sistema a través de menús, gráficos, etc, y éste le responde con resultados. También es interesante mostrar la forma en que extrae las conclusiones a partir de los hechos. En sistemas productivos se cuida la forma de presentar al operador las órdenes obtenidas del sistema experto, debido a que información excesiva o confusa dificulta la actuación en tiempo real.

En este componente se prepara la forma en la que el sistema presentará información ante el usuario. Como en los anteriores componentes nos surgen dudas y preguntas como por ejemplo:

Preguntas Comunes:

- ¿Cómo debe responder el usuario a las preguntas planteadas?
- ¿Cómo saldrán las respuestas del sistema a las preguntas que se le planteen?
- ¿Qué informaciones se representarán de forma gráfica?

La interfase de usuario debe diseñarse en base a cuatro criterios, que se considera son los más importantes y los más a tener en cuenta al desarrollar el sistema:

1.-El aprendizaje del manejo debe ser rápido.

El usuario no debe dedicar mucho tiempo aprender el manejo del sistema, debe ser intuitivo, fácil en su manejo. No debemos olvidar que nuestro sistema simula al comportamiento de un experto. Debe sernos cómodo y relativamente sencillo en cuanto al manejo.

2.-Debe evitarse en lo posible la entrada de datos errónea.

Ejemplo: pongamos en la situación de que nuestro sistema es sobre medicina. Cuando nosotros acudimos a un médico, le contamos y detallamos nuestros síntomas y el con sus preguntas junto con nuestras respuestas nos diagnostica nuestra "enfermedad". Imaginemos que acudimos a un medico y le decimos que nos duele una pierna en lugar de un brazo, el diagnostico será inútil. El ejemplo es muy exagerado pero demuestra la importancia en la correcta introducción de los datos al sistema.

3.-Los resultados deben presentarse en una forma clara para el usuario.

Volvamos al ejemplo del médico. Si nuestro médico nos diagnostica un medicamento pero en nuestra receta no nos escribe cada cuantas horas hemos de tomarlo por ejemplo, por muy bueno que sea el medicamento, la solución a nuestro problema será ineficiente por completo. Por eso se insiste en que los resultados debe ser claros y concisos

4.-Las preguntas y explicaciones deben ser comprensibles

Con estas cuatro reglas crearemos nuestra interfase con grandes posibilidades de éxito

1.3.5 COMPONENTE DE ADQUISICIÓN.

Un buen componente de adquisición ayudará considerablemente la labor del Ingeniero del Conocimiento. Este puede concentrarse principalmente en la estructuración del conocimiento sin tener que dedicar tanto tiempo en la actividad de programación. Como en el campo de la interfase, daremos unas reglas o requisitos para la realización de nuestro componente de adquisición.

Requisitos o características del componente de adquisición:

- 1- El conocimiento, es decir, las reglas, los hechos, las relaciones entre los hechos etc, deben poder introducirse de la forma más sencilla posible.
- 2- Posibilidades de representación clara de todas las informaciones contenidas en una base de conocimientos.
- 3- Comprobación automática de la sintaxis.
- 4- Posibilidad constante de acceso al lenguaje de programación.

Como se pone en práctica cada uno de los requisitos dependerá del lenguaje de programación elegido y del hardware que tengamos. El experto deberá estar algo familiarizado con el componente de adquisición para poder realizar modificaciones por sí sólo.

Existen varias formas de representar el conocimiento:

Marcos («Frames»): son estructuras de datos donde se almacenan información concreta de un cierto concepto e información relacional para completar la definición del concepto. Se representan por una serie de campos y los valores asociados a los mismos, pudiendo ser otro "frame".

Redes semánticas: son representaciones gráficas del conocimiento, mediante nodos, que representan objetos o conjuntos de objetos, y arcos, que relacionan dichos objetos. Tanto los "frames" como las redes son representaciones descriptivas, difíciles de usar para representar el razonamiento.

Reglas: es la forma más extendida de representación del conocimiento. Representan la forma de razonar. Tienen la forma *IF <condición> THEN <acció/conclusión>*. Es importante el orden de ejecución de las reglas, por ello se les suele dotar de prioridades. Una regla que activa

otra se denomina *metaregla*, y se suelen usar para desarrollar progresivamente el conocimiento del experto.

Una labor fundamental es la *Adquisición del Conocimiento* y su representación mediante alguna de las formas anteriores. Una de las técnicas más usadas son las entrevistas con los responsables del proceso, donde se consume la mayoría del tiempo. Además se suelen usar las bases de datos existentes en el sistema, añadir modelos matemáticos o heurísticas, y analizar el sistema mediante programas de simulación. También en entornos dinámicos como son los sistemas de producción, se adquiere conocimiento con el tiempo.

1.4 HERRAMIENTAS PARA EL DESARROLLO DE SISTEMAS EXPERTOS.

La implementación de un sistema experto completo obliga al desarrollo de cada uno de los elementos anteriores a través de lenguajes o herramientas de desarrollo. Existen cuatro alternativas para la implementación de un sistema experto:

" *Lenguajes de alto nivel (HLL)*: son lenguajes de propósito general (C, Fortran, Basic,...). Tienen como ventajas su eficiencia, su familiaridad y la portabilidad a cualquier entorno, pero el inconveniente de no estar preparados para una programación basada en símbolos.

" *Lenguajes simbólicos*: fundamentalmente LISP y PROLOG. Son los lenguajes de alto nivel más utilizados por su adaptación a la lógica de la base de conocimientos representada mediante símbolos, y por su mecanismo de extraer conclusiones. Su eficiencia aumenta en estaciones de trabajo diseñadas para dichos lenguajes.

" *Herramientas de desarrollo ("shells")*: son programas preparados para el desarrollo de sistemas expertos pues incorporan el motor de inferencia, independiente de las bases de hechos y de conocimiento. El programador se limita a traducir dichas bases a un lenguaje preparado a tal efecto, pudiendo enlazarse con otros lenguajes para el desarrollo de funciones adicionales. Suelen estar desarrollados en LISP o PROLOG para aumentar su eficiencia. La flexibilidad es menor, aunque suelen incorporar procesadores de texto, gráficos y herramientas de análisis.

" *Entornos de desarrollo ("environments")*: añaden a los "shells" el uso de un entorno gráfico (ventanas, iconos,...) que facilita el desarrollo del sistema experto. Tienen como ventaja la disminución del tiempo de desarrollo, pero suelen ser caros y poco flexibles.

Los "shells" son los más usados para el desarrollo de sistemas expertos, mientras que el PROLOG está muy extendido en Europa y Japón, el LISP lo está en Estados Unidos. Los entornos aún ocupan un sector minoritario pero la tendencia a su utilización va en aumento.

Como conclusiones se puede decir que los sistemas expertos están siendo muy utilizados en aquellos entornos donde se necesite el conocimiento de expertos. El entorno de la producción es uno en donde se han desarrollado más prototipos y aplicaciones de sistemas expertos. La mayoría de ellos utilizan reglas como técnica de representación, y han sido desarrollado mediante "shells", o en lenguajes simbólicos.

Una forma básica de integrar un sistema experto en el mecanismo de control es la mostrada en la siguiente figura 1.4.1, donde el sistema recoge la información en tiempo real del proceso y la procesa para dar respuesta al usuario. Los sistemas expertos parecen adecuados para la utilización como apoyo a la toma de decisiones en tiempo real, y como aprendizaje en modo estudio. La mayoría de los sistemas expertos que se utilizan en control de la producción monitorizan los resultados para que el operador tome la decisión.

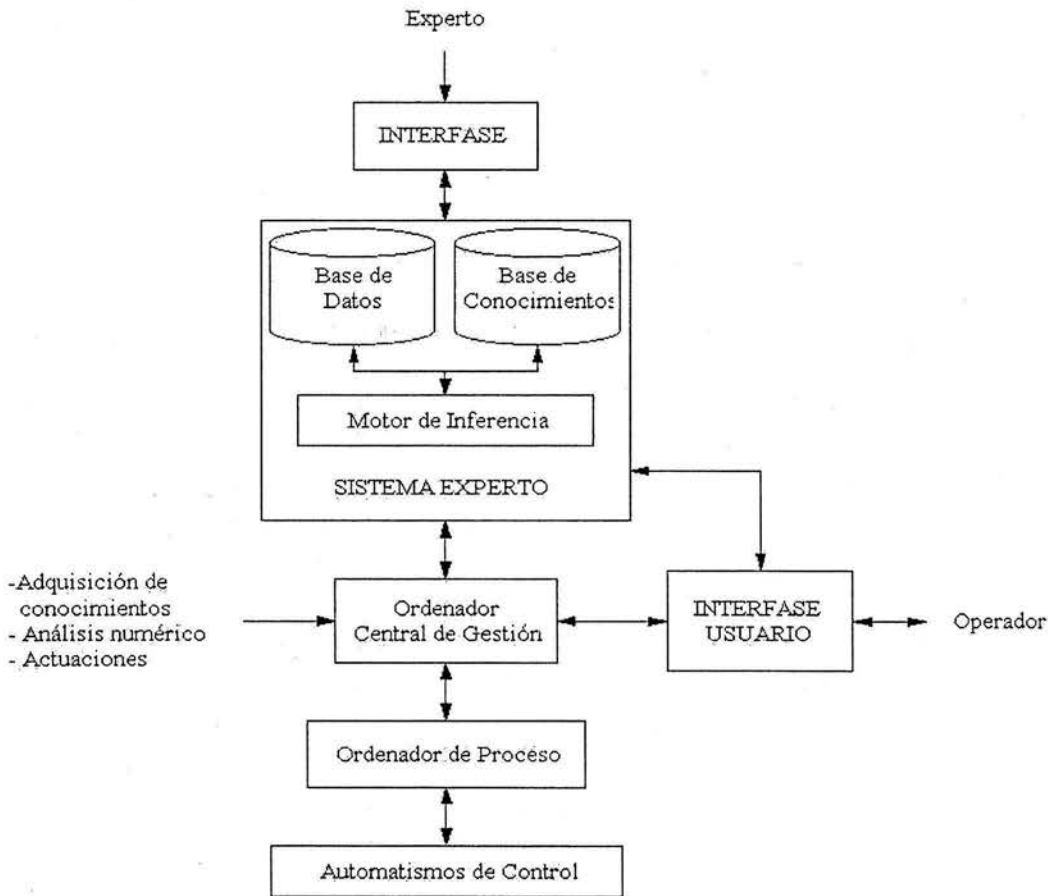


Figura 1.4.1

1.5 EL EQUIPO DE DESARROLLO.

Las personas que componen un grupo o un equipo, como en todos los ámbitos deben cumplir unas características y cada uno de ellos dentro del equipo desarrolla un papel distinto. A continuación detallaremos cada componente del equipo dentro del desarrollo y cual es la función de cada uno:

1.5.1 EL EXPERTO.

La función del experto es la de poner sus conocimientos especializados a disposición del Sistema Experto.

1.5.2 EL INGENIERO DEL CONOCIMIENTO.

El ingeniero que plantea las preguntas al experto, estructura sus conocimientos y los implementa en la base de conocimientos.

1.5.3 EL USUARIO.

El usuario aporta sus deseos y sus ideas, determinado especialmente el escenario en el que debe aplicarse el Sistema Experto.

En el desarrollo del Sistema Experto, el ingeniero del conocimiento y el experto trabajan muy unidos. El primer paso consiste en elaborar los problemas que deben ser resueltos por el sistema. Precisamente en la primera fase de un proyecto es de vital importancia determinar correctamente el ámbito estrechamente delimitado de trabajo. Aquí se incluye ya el usuario posterior, o un representante del grupo de usuarios. Para la aceptación, y en consecuencia para el éxito, es de vital y suma importancia tener en cuenta los deseos y las ideas del usuario.

Una vez delimitado el dominio, nos pondremos a "engrosar" nuestro sistema con los conocimientos del experto. El experto debe comprobar constantemente si su conocimiento ha sido transmitido de la forma más conveniente. El ingeniero del conocimiento es responsable de una implementación correcta, pero no de la exactitud del conocimiento. La responsabilidad de esta exactitud recae en el experto.

De ser posible, el experto deberá tener comprensión para los problemas que depara el procesamiento de datos. Ello facilitará mucho el trabajo. Además, no debe ignorarse nunca al usuario durante el desarrollo, para que al final se disponga de un sistema que le sea de máxima utilidad.

La estricta separación entre usuario, experto e ingeniero del conocimiento no deberá estar siempre presente. Pueden surgir situaciones en las que el experto puede ser también el usuario. Este es el caso, cuando exista un tema muy complejo cuyas relaciones e interacciones deben ser determinadas una y otra vez con un gran consumo de tiempo. De esta forma el experto puede ahorrarse trabajos repetitivos.

La separación entre experto e ingeniero del conocimiento permanece, por regla general inalterada.

1.6 VENTAJAS DE LOS SISTEMAS EXPERTOS.

El acceso al conocimiento y al juicio de un experto es extremadamente valioso en muchas ocasiones (prospecciones petrolíferas, manejo de valores bursátiles, diagnóstico de enfermedades,

etc.), sin embargo, en la mayoría de los campos de actividad existen más problemas por resolver que expertos para resolverlos. Para solucionar este desequilibrio es necesario utilizar un sistema experto. En general, actuará como ayudante para los expertos humanos y como consultor cuando no se tiene otro acceso a la experiencia.

Un sistema experto, además, mejora la productividad al resolver y decidir los problemas más rápidamente. Esto permite ahorrar tiempo y dinero. A veces sin esa rapidez las soluciones obtenidas serían inútiles.

Los valiosos conocimientos de un especialista se guardan y se difunden, de forma que, no se pierden aunque desaparezca el especialista. En los sistemas expertos se guarda la esencia de los problemas que se intenta resolver y se programa cómo aplicar los conocimientos para su resolución. Ayudan a entender cómo se aplican los conocimientos para resolver un problema. Esto es útil porque normalmente el especialista da por ciertos sus conocimientos y no analiza cómo los aplica.

Se pueden utilizar personas no especializadas para resolver problemas. Además si una persona utiliza regularmente un sistema experto aprenderá de él, y se aproximará a la capacidad del especialista.

Con un sistema experto se obtienen soluciones más fiables gracias al tratamiento automático de los datos, y más contrastadas, debido a que se suele tener informatizado el conocimiento de varios expertos.

Finalmente, la posibilidad de abordar la solución de problemas de tipo recurrente mediante un sistema experto en cualquier momento, es muy importante en las ocasiones en las que por algún motivo personal el experto humano no esté disponible.

1.7 CAMPOS DE APLICACIÓN.

Un sistema experto es muy eficaz cuando tiene que analizar una gran cantidad de información, interpretándola y proporcionando una recomendación a partir de la misma. Un ejemplo es el análisis financiero, donde se estudian las oportunidades de inversión, dependiendo de los datos financieros de un cliente y de sus propósitos.

Para detectar y reparar fallos en equipos electrónicos, se utilizan los sistemas expertos de diagnóstico y depuración, que formulan listas de preguntas con las que obtienen los datos necesarios para llegar a una conclusión. Entonces recomiendan las acciones adecuadas para corregir los problemas descubiertos. Este tipo de sistemas se utilizan también en medicina (ej. MYCIN y PUFF), y para localizar problemas en sistemas informáticos grandes y complejos.

Los sistemas expertos son buenos para predecir resultados futuros a partir del conocimiento que tienen. Los sistemas meteorológicos y de inversión en bolsa son ejemplos de utilización en este sentido. El sistema PROSPECTOR es de este tipo.

La planificación es la secuencia de acciones necesaria para lograr una meta a. Conseguir una buena planificación a largo plazo es muy difícil. Por ello, se usan sistemas expertos para

gestionar proyectos de desarrollo, planes de producción de fábricas, estrategia militar y configuración de complejos sistemas informáticos, entre otros.

Cuando se necesita controlar un proceso tomando decisiones como respuesta a su estado y no existe una solución algorítmica adecuada, es necesario usar un sistema experto. Este campo comprende el supervisar fábricas automatizadas, factorías químicas o centrales nucleares. Estos sistemas son extraordinariamente críticos porque normalmente tienen que trabajar a tiempo real.

El diseño requiere una enorme cantidad de conocimientos debido a que hay que tener en cuenta muchas especificaciones y restricciones. En este caso, el sistema experto ayuda al diseñador a completar el diseño de forma competente y dentro de los límites de costes y de tiempo. Se diseñan circuitos electrónicos, circuitos integrados, tarjetas de circuito impreso, estructuras arquitectónicas, coches, piezas mecánicas, etc.

Por último, un sistema experto puede evaluar el nivel de conocimientos y comprensión de un estudiante, y ajustar el proceso de aprendizaje de acuerdo con sus necesidades. En la tabla 1.7.1 se muestran los modelos funcionales de los sistemas expertos, junto al tipo de problema que intentan resolver y algunos de los usos concretos a que se destinan.

CATEGORÍA	TIPO DE PROBLEMA	USO
Interpretación	Deducir situaciones a partir de datos observados	Análisis de imágenes, reconocimiento del habla, inversiones financieras
Predicción	Inferir posibles consecuencias a partir de una situación	Predicción meteorológica, previsión del tráfico, evolución de la Bolsa
Diagnóstico	Deducir fallos a partir de sus efectos	Diagnóstico médico, detección de fallos en electrónica
Diseño	Configurar objetos bajo ciertas especificaciones	Diseño de circuitos, automóviles, edificios, etc
Planificación	Desarrollar planes para llegar a unas metas	Programación de proyectos e inversiones. Planificación militar
Monitorización o supervisión	Controlar situaciones donde hay planes vulnerables	Control de centrales nucleares y factorías químicas
Depuración	Prescribir remedios para funcionamientos erróneos	Desarrollo de software y circuitos electrónicos
Reparación	Efectuar lo necesario para hacer una corrección	Reparar sistemas informáticos, automóviles, etc
Instrucción	Diagnóstico, depuración y corrección de una conducta	Corrección de errores, enseñanza

Control	Mantener un sistema por un camino previamente trazado. Interpreta, predice y supervisa su conducta	Estrategia militar, control de tráfico aéreo
Enseñanza	Recoger el conocimiento y mostrarlo	Aprendizaje de experiencia

Tabla 1.7.1

CAPITULO 2

CONTROL ESTADÍSTICO DE CALIDAD

El control de calidad de los productos manufacturados es una función que existía desde hace mucho tiempo antes de la aplicación de los métodos estadísticos en el análisis de los datos de calidad y que todavía está en uso, sin importar que se empleen o no las técnicas estadísticas. Gran parte de la estadística se basa en la ley de los grandes números y en la teoría matemática de las probabilidades. Si la expresión control de calidad se emplea en la forma correcta, toma una función mucho más amplia que la expresión control estadístico de calidad.

La expresión control de calidad se refiere al control de calidad del producto con cualquier método utilizado y es el adjetivo estadístico el que se refiere al control de calidad del producto por métodos estadísticos.

2.1 CONTROL ESTADÍSTICO DEL PROCESO.

Una herramienta importante en el control estadístico de calidad es la gráfica de control de Shewhart cuyo nombre se debe a que la técnica la estableció el doctor Walter A. Shewhart en la década de 1920 cuando estaba con Bell Telephone Laboratories. La gráfica de control tiene su fundamento en la teoría siguiente:

La calidad medida de un producto manufacturado siempre esté sujeta a cierta cantidad de variación como resultado de la casualidad. “Algún sistema de causas ocasionales estable es inherente a cualquier método particular de producción e inspección. La variación, dentro de este patrón estable, es inevitable. Las razones de la variación ajenas a este patrón estable se pueden descubrir y corregir”.

La fuerza de la técnica de Shewhart radica en su capacidad para separar estas causas asignables de la variación en la calidad.

Al determinar que algunas de las variaciones en la calidad son variaciones ocasionales, la grafica de control señala cuando no hay que intervenir más en un proceso, lo cual evita ajustes innecesarios que tienden a incrementar la variabilidad del proceso en vez de reducirla.

2.2 MUESTREO PARA ACEPTACIÓN.

El muestreo para aceptación es un de los campo más amplios del control estadístico de calidad. Una compañía que recibe un embarque, lo muestrea para aceptarlo o rechazarlo, de acuerdo con las normas, Si es rechazado por encontrarse debajo de las normas, puede ser devuelto o conservado, dependiendo de la necesidad que se tenga de la mercancía, o de los acuerdos que se hayan tenido con el proveedor.

Con frecuencia la propia producción de una compañía es sometida a un muestreo de aceptación en varias etapas de la producción. Un lote de un determinado producto es muestreado, y si es aceptado para que continúe el proceso, se le envía a los clientes, o bien es rechazado si no cumple con las normas establecidas.

El muestreo de aceptación determina un procedimiento que si se aplica a una serie de lotes dará un riesgo especificado en cuanto a la aceptación de lotes de una calidad dada. Puede decirse que, el muestreo de aceptación da un margen de seguridad en cuanto a la calidad.

Dicho de otra forma, el muestreo de aceptación no "controla" la calidad. Como lo es el propósito de las graficas de control, estas guían al ingeniero en cuanto a la modificación de la producción, con el objeto de generar mejores productos. Este es el verdadero control estadístico de calidad. Un muestreo de aceptación simplemente acepta o rechaza los lotes. Si los lotes son todos de la misma calidad, aceptará unos y rechazará otros, y los aceptados no necesariamente serán de mejor calidad que los rechazados.

Si un fabricante utiliza el muestreo de aceptación en varias etapas de la producción, puede tener efectos favorables con respecto a la calidad de la producción. Por ejemplo si una compañía solo utiliza un muestreo para la inspección del producto terminado(inspección final) y descuida en esta inspección final las mercancías enviadas a los clientes, puede darse una actitud desobligada en el personal de producción en cuanto a la calidad. Esto es porque los trabajadores pueden considerar que no será embarcado un producto de mala calidad, ya que seria detectado y rechazado en la inspección final. La responsabilidad del departamento de producción entonces seria, la cantidad de producción más que la calidad. Además que las inspecciones finales puede que no sean tan eficaces; por su carácter monótono y aburrido, un buen porcentaje de productos defectuosos podrá pasar la inspección final, mientras mayor sea la cantidad, entonces menor será la calidad de la inspección. Por otra parte, con un programa de muestreo de aceptación, el costo de vigilar y repetir el trabajo puede ser cargado al departamento de producción. El personal de producción se hará consiente de la calidad, y el departamento de inspección como el departamento de producción tendrán interés en esta. Una buena actitud por parte del departamento de producción seria: "hagámoslo bien desde la primera vez".Los aspectos psicológicos del muestreo de aceptación tienen una gran importancia.

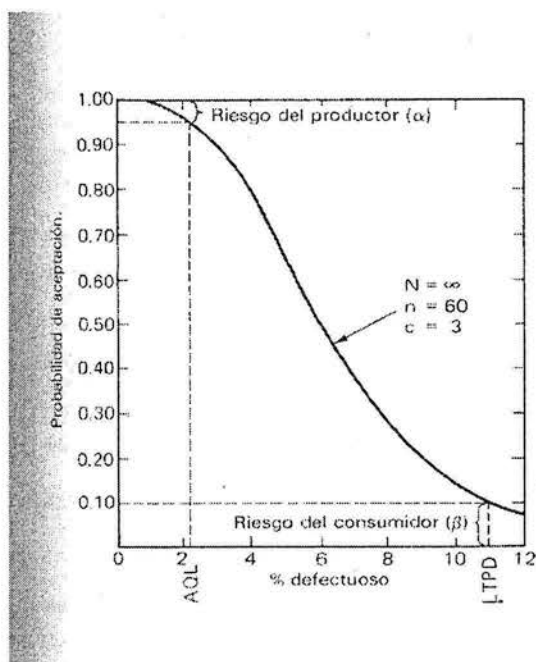
El muestreo de aceptación podría ser utilizado convenientemente en los puntos siguientes:

1. Cuando el costo de inspección es elevado, y las pérdidas que son consecuencia de una unidad defectuosa no son demasiado grandes. Es posible que el proceso más económico sea, el de no realizar inspección alguna.
2. Cuando una inspección al 100 por ciento resulta exhaustiva, un proceso de muestreo bien preparado producirá buenos resultados. Como se dijo antes, la inspección final puede no significar una calidad perfecta y el porcentaje de elementos defectuosos podrá ser mas elevado que cuando se aplica un proceso de muestreo científicamente preparado.
3. Cuando la inspección es destructiva. En este caso deberá emplearse el muestreo.

2.2.1 LA CURVA OC.

La curva OC representa los riesgos inherentes a un plan de muestreo particular. En un lote con un porcentaje defectuoso especificado, la curva da la probabilidad de que el lote resulte aceptado por el plan de muestreo. Cada plan tiene su curva OC particular.

En la figura 2.2.1.1(1) se muestra una curva típica OC.



p'	np'	P_a
0.00	0.00	1.000
0.01	0.60	0.997
0.02	1.20	0.966
0.03	1.80	0.891
0.04	2.40	0.779
0.05	3.00	0.647
0.06	3.60	0.515
0.07	4.20	0.395
0.08	4.80	0.294
0.09	5.40	0.213
0.10	6.00	0.151
0.11	6.60	0.105
0.12	7.20	0.072

Figura y Tabla 2.2.1.1(1)

Cada curva empieza con un porcentaje de aceptación igual al 100% y un porcentaje defectuoso, igual a cero. Conforme aumenta al porcentaje defectuoso disminuye la tasa de aceptación. El declive de la curva mide la capacidad del plan para diferenciar entre diversos niveles de calidad y relacionarlos al costo y cantidad de inspección.

El cálculo de las probabilidades que se muestra en la ordenada de la figura 2.2.1.1(1), requiere el empleo de la distribución binomial o de la de Poisson, ya que la estadística (porcentaje defectuoso) se derivó de datos discretos.

En la figura 2.2.1.1(1), la medida del lote (N) es ∞ (infinito), la medida de la muestra (n) es 60, y el número de aceptación (c) es 3. En la determinación de las probabilidades solamente son necesarios la medida de la muestra y el número de aceptación. Como $c = 3$, cualquier muestra que contenga 0,1,2 ó 3 unidades defectuosas, debe dar por resultado la aceptación del lote. El intervalo de valores del porcentaje defectuoso pueden ir desde cero hasta cualquier clase de valor para el porcentaje defectuoso de la calidad del lote. En este caso, el intervalo varía de 0 a 12. Para entrar a las tablas de Poisson es necesario conocer los valores np' , como se ilustra en la tabla 2.2.1.1(1).

En la figura 2.2.1.1(1), las curvas OC quedan definidas por dos niveles de calidad. El primero está dado por AQL (nivel de calidad aceptable) y el segundo por LTPD (tolerancia de porcentaje defectuoso en el lote). Donde AQL es el nivel de calidad deseable, en tanto que LTPD es un nivel de calidad no deseable.

El nivel AQL está definido como el material que se considera como “bueno”. Cuando los lotes que contienen este porcentaje defectuoso quedan sometidos a una inspección, la aceptación debe ser relativamente “alta”. Esta “aceptación alta”, se localiza alrededor del 95%, aun cuando puede variar entre 90 y 99%, dependiendo del plan.

Como se supone que para el consumidor es aceptable el valor AQL, el productor está interesado en la probabilidad de que se rechacen los lotes que contienen este porcentaje defectuoso. Lo anterior se ilustra en la figura 2.2.1.1(1), en la referencia riesgo del productor (α). Los riesgos del productor son el complemento de la aceptación de los consumidores al nivel AQL. Todos los planes de muestreo incluidos en el MIL-STD-105D quedan definidos por el valor AQL.

El nivel LTPD está definido como el material que se considera como “malo”. Cuando se someten a una inspección los lotes que contienen este porcentaje defectuoso, la aceptación debe ser relativamente “baja”. Esta “aceptación baja” esta definida como el 10%, ya que el consumidor desea minimizar los riesgos de que se acepten los lotes malos. En la figura 2.2.1.1(1) También se ilustran los riesgos del consumidor (β) en conexión con el nivel LTPD.

Con frecuencia la selección del valor correcto de AQL o LTPD es una decisión arbitraria tomada por un ingeniero de control de calidad o de alguna otra persona durante la inspección. Algunas otras áreas administrativas también participan para especificar el porcentaje defectuoso apropiado para una aplicación específica.

Los dos niveles de calidad AQL y LTPD dividen el área bajo la curva en tres zonas.

1. Lotes buenos (AQL o mínimo porcentaje defectuoso), los cuales tendrán aproximadamente un porcentaje de aceptación igual o mayor al 95%.
2. Lotes intermedios (porcentaje defectuoso entre AQL Y LTPD), los cuales tendrán varias probabilidades de aceptación.
3. Lotes malos (LTPD o porcentaje defectuoso mayor), los cuales tendrán aproximadamente una aceptación igual o menor al 10%.

El ancho de la zona intermedia mide la capacidad de discriminación del plan de muestreo. Una zona ancha da por resultado muestras pequeñas e inspecciones económicas pero incrementa el riesgo total. Una zona angosta da por resultado muestras grandes, incrementos en la inspección y disminución del riesgo. Es necesario, tener conocimientos estadísticos e información especializada para poder equilibrar el costo de la inspección con los riesgos.

Los niveles de calidad, AQL Y LTPD, y sus riesgos asociados, α y β , limitan la selección del plan de muestreo. El designar solo un punto sobre la curva OC (AQL y α o LTPD y β) se amplía la selección del plan de muestreo, porque existen muchos planes que pueden satisfacer un solo criterio. La figura 2.2.1.2(1) ilustra lo que sucede cuando AQL y α especifican el punto sobre la curva OC. Conforme disminuye la medida de la muestra la protección contra la aceptación de lotes de baja calidad disminuye. La figura 2.2.1.3(1) ilustra lo que sucede cuando LTPD y β especifican el punto sobre la curva OC. Conforme disminuye la medida de la muestra, menos lotes buenos son aceptados, lo que reduce los costos del muestreo, pero se crean problemas con el proveedor debido a que resultarán rechazados más lotes buenos.

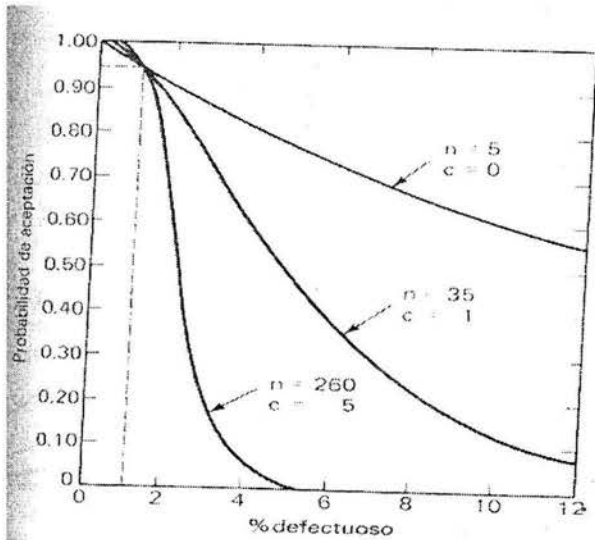


Figura 2.2.1.2(1)

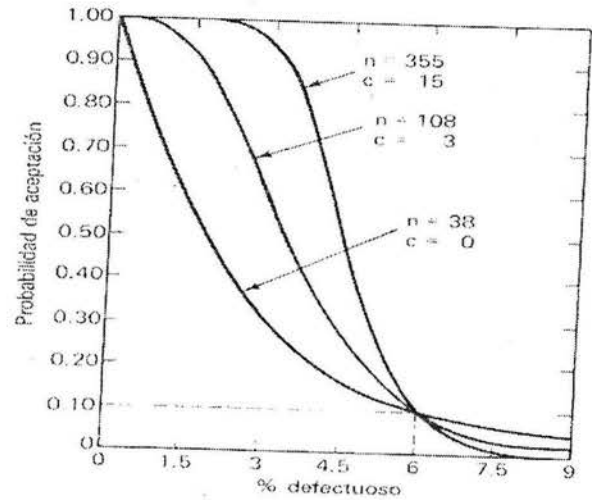


Figura 2.2.1.3(1)

Lo anterior ilustra como operan los planes de muestreo a largo plazo. Se toma de un lote una muestra de tamaño específico y si no excede el número de aceptación, se acepta el uso de dicho lote. Pero cuando se rechaza un lote, se tienen dos alternativas:

- regresar el material al proveedor
- conservar el material y revisarlo al 100%.

NOTA: La primera acción es costosa, ya que se requiere exceso de transportación que puede resultar en un paro en la producción debido a la inestabilidad de los inventarios limitados. Cuando se requieren partes, la única opción factible puede ser la clasificación. Esto nos conduce a otro concepto de muestreo, llamado AOQ (calidad promedio de salida), que se utiliza en algunas ocasiones. En la figura 2.2.1.5(1) siguiente se ilustra este concepto.

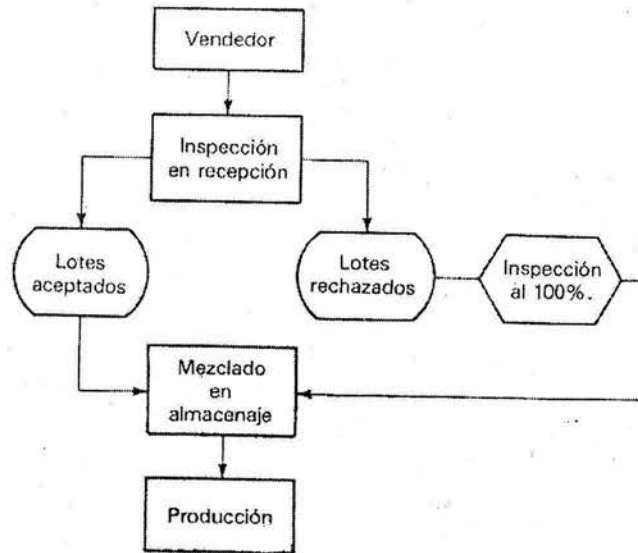


Figura2.2.1.5(1)

2.2.2 PLANES DE MUESTREO DE ATRIBUTOS.

Los planes de muestreo de características o atributos son las técnicas de muestreo que se emplean con mas frecuencia debido primordialmente a que son fáciles de administrar y las tablas impresas se encuentran disponibles con facilidad.

Las dos mayores fuentes de planes de muestreo de características son:

1. Tablas de inspección de muestreo, muestreo simple y doble, por H.F Dodge y H.G Roming (conocidas comúnmente como las tablas de Dodge-Romig). Estas tablas enlistan los planes de muestreo para dos clasificaciones:
 - a) Planes LTPD: estos varían de 0.5 a 10%. Los procesos promedio varían de cero a una y media veces los valores LTPD especificados. El valor de β se fija en 10%. Se propone una inspección reducida o mínima.
 - b) Planes AOQL: que varían de 0.1 a 10.0%. los procesos promedio varían de cero al valor AOQL. El valor de β se fija en 10%. Se propone una inspección reducida o mínima.
2. Los procedimientos militares de muestreo estándar y las tablas para inspección por características (conocidas comúnmente como MIL-STD-105D).

- a) Los procedimientos y tablas MIL-STD-105D, asientan los planes de muestreo simple, doble y múltiple caracterizado por los valores AQL y diversos niveles y grados de inspección.
- b) Los valores de AQL varían de 0.015 a 10%. El valor de α varia de 20% (muestra pequeña) hasta 0.2% (muestra grande).
- c) La inspección de muestra pequeña tiene cuatro niveles de inspección.
- d) La inspección ordinaria tiene tres niveles de inspección.
- e) Los grados de inspección son: normal, rigurosa y reducida.

2.2.2.1 VARIABLES Y ATRIBUTOS.

Una distinción importante en el lenguaje técnico de la estadística es la que debe hacerse entre variables y atributos. Cuando se hace el registro de una característica medida de la calidad, tal como una dimensión expresada en milésimas de pulgada, se dice que la calidad está expresada por variables. Cuando en un registro aparece solo el número de artículos que cumplen y el número de los que no cumplen con cualesquiera requisitos especificados, se dice que es un registro por atributos.

Todos los productos manufacturados deben cumplir con ciertos requisitos, sean expresos o implícitos. Muchos de estos requisitos se pueden expresar como variables. Los ejemplos son las dimensiones, las temperaturas de calentamiento en grados Fahrenheit o Celsius, el peso en libras o kilogramos del contenido de cualquier recipiente, el tiempo en segundos hasta que se funde un fusible, la duración (vida) en horas de una lámpara. En casi todas las especificaciones de variables se dan los límites superior e inferior del valor medido. En algunas, tales como el porcentaje de una impureza particular en un compuesto químico puede tener solo un límite superior, mientras que otro, como la resistencia física, puede tener solo un límite inferior. Las variables que se manejan en las gráficas de control de Shewhart son \bar{X} y R y también \bar{X} y s .

Muchos requisitos, por necesidad, se expresan en términos de atributos en vez de variables. Ello se aplica, en muchas cosas que solo se pueden juzgar por inspección visual por ejemplo: La tapa de vidrio de un recipiente agrietada, una etiqueta tiene o no cierto color deseado o el acabado de la superficie de un mueble presenta o no aspecto satisfactorio. En general, la cosa examinada cumple o no cumple con las especificaciones.

2.2.2.2 TIPOS DE MUESTREO EN LOS PLANES DE MUESTREO POR ATRIBUTOS.

En la elección de un plan de muestreo existen tres diferentes tipos a considerar, estos quedan descritos por el número de muestras que deben tomarse de un lote antes de tomar la decisión de aceptarlo o rechazarlo.

- *Muestreo único.* Se selecciona al azar una muestra (n) de un lote (N) y se inspecciona. Se cuentan las unidades (d) que no cumplan con los requisitos, y se comparan con el número de aceptación (c) del plan. Si d es menor o igual que c se acepta el lote. Si d es mayor que c , se rechaza el lote.
- *Doble muestreo.* Bajo este tipo de muestreo, cuando el resultado del estudio de la primera muestra no es decisivo, una segunda muestra es extraída de la misma población. Las dos

muestras son combinadas para analizar los resultados. Este método permite a una persona principiar con una muestra relativamente pequeña para ahorrar costos y tiempo. Si la primera muestra arroja un resultado definitivo, la segunda muestra puede no necesitarse. Por ejemplo, al probar la calidad de un lote de productos manufacturados, si la primera muestra arroja una calidad muy alta, el lote es aceptado; si arroja una calidad muy pobre, el lote es rechazado. Solamente si la primera muestra arroja una calidad intermedia, será requerirá la segunda muestra. Un plan típico de muestreo doble puede ser obtenido de la Military Standard Sampling Procedures and Tables for Inspection by Attributes, publicada por el Departamento de Defensa de EE.UU y también usado por muchas industrias privadas. Al probar la calidad de un lote consistente de 3,000 unidades manufacturadas, cuando el número de defectos encontrados en la primera muestra de 80 unidades es de 5 o menos, el lote es considerado bueno y es aceptado; si el número de defectos es 9 o más, el lote es considerado pobre y es rechazado; si el número está entre 5 y 9, no puede llegarse a una decisión y una segunda muestra de 80 unidades es extraída del lote. Si el número de defectos en las dos muestras combinadas (incluyendo $80 + 80 = 160$ unidades) es 12 o menos, el lote es aceptado si el número combinado es 13 o más, el lote es rechazado.

- *Muestreo múltiple.* Este plan extiende el concepto descrito en el punto anterior a una mayor cantidad posible de muestreos, tantos como nueve, antes de tomar una decisión. En cada paso (muestreo), se acumulan y cuentan las unidades que no cumplen con los requisitos y se comparan con el número de aceptación de cada muestra. En cada paso es posible adoptar cualquiera de las siguientes decisiones, aceptar, continuar o rechazar. Sin embargo, en la última muestra sólo se puede tomar una decisión de aceptación o rechazo.

NOTA: Cuando estos planes se encuentran en las tablas formales de muestreo, se diseñan para que tengan la misma curva OC. Por lo que se obtiene la misma protección con el muestreo único, doble o múltiple. La selección de un tipo particular de muestreo depende entonces de otros factores como los siguientes.

- Enseñar el plan a los inspectores y vendedores
- Supervisión del plan.
- Extracción de la muestra
- Espacio de almacenamiento
- Establecimiento de un inventario

El número de muestra promedio que se necesita para llegar a una decisión, es menor en los muestreos doble y múltiple que en el muestreo simple. La ventaja de esto es un menor costo en la inspección, la desventaja es la dificultad para encontrar quién enseñe los planes al personal de la inspección.

2.2.3 DATOS HISTÓRICOS DE LOS SISTEMAS DE MUESTREO PARA ACEPTACIÓN BASADOS EN EL CONCEPTO DE AQL.

Los procedimientos militares estándares de muestreo para inspección por atributos se crearon durante la segunda guerra mundial. Los procedimientos y tablas para pertrechos militares se prepararon en 1942, y pasaron después a convertirse (con modificaciones) en tablas para las fuerzas del Servicio del Ejército de EEUU. Las tablas de la Armada de EEUU fueron publicadas en 1945 y adoptadas en 1949 como norma conjunta Ejército-Armada. Este JAN-STD fue sustituido por el Mil.std.105A en 1950. Desde entonces, ha habido modificaciones menores en la norma 105, publicada como 105B y 105C. La Military Standard 105D que se emplea en esta tesis es el resultado de un estudio llevado a cabo por el grupo de trabajo americano-británico-canadiense que trató de lograr establecer un patrón común para los tres países. Mil.Std.105D fue publicada por el gobierno de Estados Unidos en 1963. En 1971 fue adoptada por el American National Standards Institute como ANSI Standards Z1.4, y en 1973, salvo por pequeños cambios editoriales, fue adoptada por la Internacional Organization for Standardization como ISO 2859.

2.2.4 DESCRIPCIÓN DE LA NORMA MILITAR ESTÁNDAR (MIL-STD-105D).

Los procedimientos de muestreo mencionados anteriormente son casos individuales, que reúnen ciertas características, como por ejemplo p_1' , p_2' , con α y β dados. La Mil.Std.105D es un conjunto de esos procesos organizados en un sistema de métodos de muestreo. Un método (scheme) de muestreo consiste en una combinación de un proceso de muestreo normal, uno de muestreo severo o riguroso y uno de muestreo reducido o abreviado, con las reglas para cambiar de uno a otro. Puede tener también cláusulas para discontinuar la inspección si un número especificado de lotes consecutivos permanece en inspección severa o rigurosa.

El punto principal de la Mil.Std.105D es especificar el nivel aceptable de calidad o AQL para una determinada característica de calidad. Se espera que el proveedor entonces someta a una inspección varios lotes de un producto, y el propósito del procedimiento de muestreo Mil.Std.105D es, forzar al proveedor para que produzca ese producto con la calidad AQL deseada. Esto se logra no solamente mediante la aceptación y rechazo de lotes en un proceso de muestreo particular, sino al cambiar a otro procedimiento de muestreo más exigente, en cuanto se deteriore el producto del proveedor, pasando a ser inferior al AQL convenido.

La Mil.Std.105D da una serie de AQLs. Para los procesos de fracción defectuosa, el AQL varía desde 0.10 a 10 por ciento. Para procedimientos de defectos por unidad hay 10 AQLs adicionales que llegan hasta mil defectos por cada cien unidades. Así que para los niveles más bajos de AQL, el mismo proceso de muestreo puede ser utilizado para controlar una fracción defectuosa o el número de defectos por unidad.

Se debe tomar una decisión acerca de un AQL, y también es necesario, cuando se aplica la Mil.Std.105D, decidir acerca del "nivel de inspección". Esto determina la relación entre los tamaños del lote y la muestra. Se ofrecen tres niveles generales de inspección. El nivel II es el que se considera como normal. el nivel I puede especificarse cuando se necesita menos selección, y el nivel III cuando se requiere una diferencia mayor. Hay también cuatro niveles especiales. La decisión acerca de cual nivel de inspección se usará, se basa en el tipo de producto del que se trate. Para elementos poco costosos y sencillos, el nivel puede ser bajo; para piezas costosas y

complejas deberá ser alto. Si la inspección es perjudicial para el producto, puede utilizarse un bajo nivel de inspección. El nivel de inspección se adopta al iniciar el programa de muestreo, y no se le cambia, a partir de ese momento.

Para un AQL especificado, un nivel de inspección y un tamaño de lote dados, el Mil.Std.105D da un proceso de muestreo normal, que se utiliza siempre que el proveedor esté generando productos que tengan la calidad AQL o mejor. Pero también ofrece un proceso más severo al cual cambia si hay pruebas de que la calidad se ha deteriorado. Una regla marca el cambio del proceso normal al más exigente (rigurosa). Este cambio se realiza después de que dos de cinco lotes consecutivos han sido rechazados durante la inspección original (normal). Se vuelve a implantar la inspección normal, cuando cinco lotes consecutivos han sido aceptados en el curso de la inspección original (rigurosa). Si se encuentran 10 lotes consecutivos en el proceso más estricto, se detiene la inspección (se discontinúa) para proceder como se necesite o convenga con respecto a la calidad. El cambio de una inspección más estricta a la normal y viceversa, se refiere a un proveedor en particular, y es independiente de los niveles de inspección originalmente adoptados, los cuales se refieren al tipo de producto y no al proveedor.

La Mil.Std.105D ofrece tres tipos de procedimientos de muestreo. Los tres tipos se refieren a procesos de muestreo sencillo, doble y múltiple. La selección entre uno y otro se hace según convenga administrativamente.

Debido a la posibilidad de escoger entre los tres tipos de procesos, el estándar no muestra inmediatamente el tamaño de la muestra, sino que suministra inicialmente una letra código para el tamaño. Esto, junto con la decisión del tipo de procedimientos, señalan el proceso de muestreo específico por utilizar.

Los pasos en el uso de las normas pueden resumirse como sigue:

1. Decisión del AQL.
2. Decisión del nivel de inspección
3. Determinación del tamaño del lote
4. Consultar la tabla para encontrar la letra código correspondiente al tamaño de la muestra
5. Decisión en cuanto al procedimiento de muestreo a utilizar
6. Uso de la tabla para encontrar el proceso por utilizar
7. Uso de la tabla de inspección más estricta, para el mismo tipo de proceso, cuando se requiera cambiar a una inspección más estricta.

Como un auxiliar en los procedimientos de muestreo anteriores, la Mil.Std.105D tiene cambios a inspección reducida si la calidad resulta especialmente buena. Una regla marca el cambio del proceso normal al abreviado (reducida). La producción debe seguir un ritmo normal (o ser estable), se deben haber aceptado los 10 últimos lotes durante la inspección normal, y el número total de defectuosos (o defectos) debe ser menor al valor estipulado en la tabla correspondiente. La idea básica de la inspección abreviada o reducida, es ahorrar dinero cuando la calidad esta resultando buena.

La norma tiene tablas suplementarias que abarcan los puntos del 10 y 5 por ciento de las curvas CO. Estos valores se dan para limitar el riesgo de aceptación de lotes individuales peores que una fracción defectuosa de la tolerancia del lote, o cierto número de defectos por unidad en la tolerancia del lote. Hay además una tabla de valores AOQL para procedimientos de inspección normal y estricta.

2.2.4.1 USO DE LA NORMA MILITAR ESTÁNDAR.

Estas tablas tienen las medidas adecuadas para clasificar un defecto como crítico, mayor o menor, empleando en cada clasificación los diferentes valores de AQL. A continuación se resume la información requerida para su empleo:

1. De los valores dados selecciónese el valor de AQL que se va a utilizar. Los valores de AQL menores o iguales a 10.0, pueden expresarse como porcentaje defectuoso o (defectos por cada cien unidades). Los mayores de 10.0 únicamente se expresan como defecto por cada cien unidades. Por ejemplo: se tiene un valor de AQL dado igual a 2.5%.
2. Se debe determinar el tamaño del lote. Véase la tabla I, para encontrar la letra codificada correspondiente a la medida de la muestra. Usualmente está especificado el nivel II de inspección general. El nivel I es el discriminante menor, y el nivel III es el discriminante mayor. Cuando es necesario trabajar con muestras pequeñas y pueden tolerarse grandes riesgos, se utilizan niveles de inspección especiales. Por ejemplo: El tamaño del lote es de 1000 piezas, el nivel de inspección especificado es II, en la tabla I, la letra codificada correspondiente es J.
3. Selecciónese el plan de muestreo adecuado.
 - a) Empléese la tabla II-A para muestreo sencillo. Por ejemplo: Letra codificada es J, la medida de la muestra es entonces 80. Con un valor AQL igual a 2.5% se aceptan los lotes con cinco defectos y se rechazan los lotes con seis defectos en adelanté.

Las tablas I, II-A se presentan en los anexos.

2.2.4.2 TEXTO DE LA NORMA MILITAR ESTÁNDAR.

Nota acerca del significado del AQL

Su aceptación, (la del consumidor) mediante un proceso de muestreo dará por buena la mayoría de los lotes o conjuntos que el proveedor someta, con tal que el nivel promedio del proceso, en cuanto al por ciento defectuoso (o defectos por cada cien unidades) en dichos lotes o conjuntos, no sea mayor que el valor asignado al AQL. En esta forma, el AQL es un valor asignado del por ciento defectuoso (o defectos por cada cien unidades) que el consumidor indica será aceptado por el procedimiento de muestreo a utilizar la mayor parte del tiempo. Los procedimientos de muestreo que se analizan, se disponen en forma tal, que la probabilidad de aceptación para el mayor valor asignado de AQL, depende del tamaño de la muestra, siendo en general mayor para muestras grandes que para pequeñas, para un

determinado AQL. El AQL, por sí mismo, no representa protección para el consumidor en lotes o conjuntos individuales, sino que se relaciona más directamente con lo que puede esperarse de una serie de lotes o conjuntos, siempre y cuando los pasos indicados sean los que se siguen. Es necesario referirse a la curva característica de operación del proceso para determinar cuál será la protección que alcance el consumidor.

Especificación del AQL

El AQL a utilizar debe especificarse en el contrato, o por la autoridad responsable. Pueden especificarse diferentes AQLs para conjuntos de defectos considerados colectivamente o para defectos individuales. Puede especificarse un AQL para un conjunto de defectos además del AQL para defectos individuales o subconjuntos, dentro de dicho conjunto. Los valores de AQL de 10.00 menos pueden expresarse ya sea en por ciento defectuoso, o en defectos por cada cien unidades; los que superan el 10.0 deberán ser expresados en defectos por cada cien unidades únicamente.

Lote o conjunto

La expresión lote o conjunto significara "lote para inspección" o "conjunto para inspección", es decir, una colección de unidades del producto, de la cual se sacara una muestra y se la inspeccionara para determinar su conformidad con los criterios de aceptabilidad, y pueden diferir de los de una colección de unidades designadas como lote o conjunto para otros propósitos (por ejemplo, producción, embarque, etc.)

Formación de lotes o conjuntos

El producto debe ser reunido en lotes identificables, sublotes, conjuntos, o de cualquier otra manera que se prescriba. Cada lote o conjunto deberá, en cuanto sea practicable, constar de unidades del producto de un solo tipo, grado, clase, tamaño y composición, fabricado en consideraciones esencialmente iguales y al mismo tiempo.

Tamaño del lote o conjunto

El tamaño del lote o conjunto es el número de unidades del producto que forman el lote o conjunto.

Muestra

Una muestra esta formada por uno o más elementos del producto, tomadas del lote o conjunto, efectuándose esta selección en forma al azar y sin tomar en cuenta su calidad. El número de elementos del producto en la muestra constituye el tamaño de la misma.

Proceso de muestreo

Un proceso de muestreo indica el número de elementos del producto de cada lote o conjunto que habrán de ser inspeccionado (tamaño de la muestra o series de tamaños de la muestra) y los criterios para determinar la aceptabilidad del lote o conjunto (números de aceptación y rechazo).

Proceso de muestreo sencillo

El número de elementos de muestra inspeccionados, deberá ser igual al tamaño de la muestra dada por el proceso. Si el número de elementos defectuosos encontrado en la muestra es igual o menor que el número de aceptación, el lote o conjunto será considerado aceptable.

Si el número de elementos defectuosos es igual o mayor que el número de rechazo, el lote o conjunto será rechazado.

Inspección del porcentaje defectuoso

Para determinar la aceptabilidad de un lote o conjunto de acuerdo con la inspección de porcentaje defectuoso, se utilizarán los procedimientos aplicables de muestreo de acuerdo con el proceso de muestreo sencillo.

2.2.5 CRITERIOS PARA EL CAMBIO DE INSPECCIÓN NORMAL, RIGUROSA Y REDUCIDA.

- **Inicio de la inspección**

Al principiar la inspección se usará la inspección normal a menos que la autoridad responsable disponga otra cosa.

- **Continuación de la inspección.**

La inspección normal, más severa, o reducida, continuará sin cambio para cada clase de defectos o elementos defectuosos en lotes o conjuntos sucesivos, excepto cuando los procedimientos para cambios que se dan adelante, requieran dicho cambio. Los procedimientos para cambios se aplicarán a cada clase de defectos o elementos defectuosos en forma independiente.

Procedimientos de cambio.

- **Normal a rigurosa.** Cuando se está haciendo una inspección normal, la inspección rigurosa deberá aplicarse cuando dos de cinco lotes o conjuntos consecutivos hayan sido rechazados en la inspección original (es decir, ignorando los lotes o conjuntos que se hayan vuelto a examinar con este procedimiento).
- **De rigurosa a normal.** Cuando se lleva a cabo una inspección severa, podrá regresarse la inspección normal, cuando cinco lotes o conjuntos consecutivos han sido considerados aceptables en la inspección original.
- **De normal a reducida.** Cuando se efectúa una inspección normal, puede cambiarse a inspección reducida, siempre y cuando se satisfagan todas las condiciones siguientes.
 - a) Los 10 lotes o conjuntos precedentes o más, han pasado la inspección normal original y ninguno ha sido rechazado;
 - b) El número total de elementos defectuosos (o defectos) en las muestras de los 10 lotes o conjuntos precedentes es igual o menor que el número aplicable dado por la tabla VIII. Si se utiliza muestreo doble o múltiple, todas las muestras inspeccionadas deberán ser incluidas, no sólo las "primeras" muestras;
 - c) La producción tiene un ritmo constante o estable; y
 - d) La inspección reducida es considerada apropiada por la autoridad competente.

- **Reducida a Normal.** Cuando se utiliza la inspección reducida, deberá cambiarse a inspección normal si durante la inspección original se presenta las siguientes condiciones:
 - a) Se rechaza un lote o conjunto;
 - b) Un lote o conjunto se considera aceptable pero los defectos (rechazos) encontrados estuvieron entre A_c y R_e del plan.
 - c) La producción se hace irregular o retrasada; u
 - d) Otras condiciones aconsejan que se reanude la inspección normal.

- **Interrupción de la inspección.** En el caso de que 10 lotes o conjuntos consecutivos continúen bajo la inspección más rigurosa (ú cualquier otro número que señale la autoridad responsable), la inspección de acuerdo con los requisitos de este documento, deberá ser suspendida en espera de que se mejore la calidad del material sometido a inspección.

Lo anterior se muestra en la figura 2.2.5.1(2)

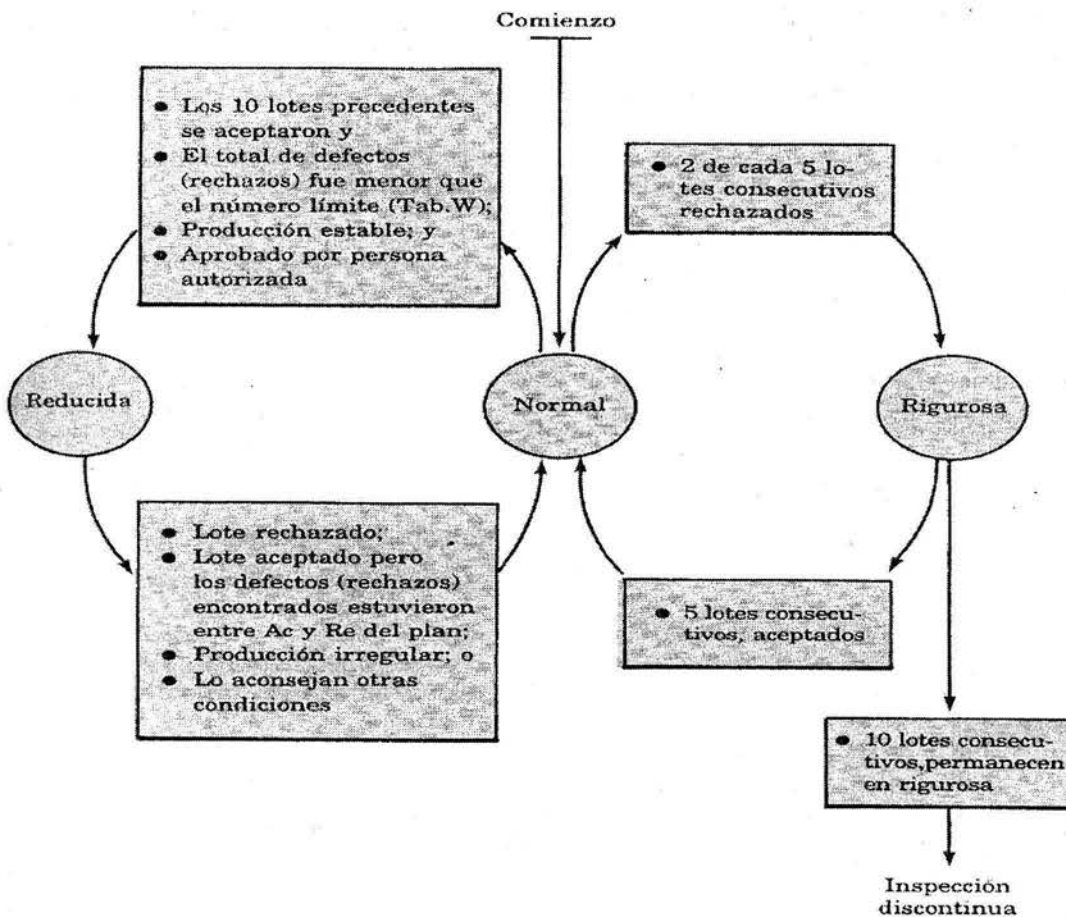


Figura 2.2.5.1(2)

2.2.5.1 REGLAS DE CAMBIO DE INSPECCIÓN NORMAL, RIGUROSA Y REDUCIDA EN EL SHELL I2+.

En la figura 2.2.5.1(2) anterior se ilustran y se describen los requerimientos necesarios para el cambio de los procedimientos de muestreo de normal a severo o riguroso y viceversa o de normal a reducido o abreviado y viceversa o discontinuación de la inspección, estos pasos descritos teóricamente pueden ser redactados como una serie de reglas que indiquen el cambio a los diferentes tipos de inspecciones (normal, rigurosa, reducida).

Debido a que precisamente los sistemas expertos necesitan del conocimiento de los expertos expresado en reglas de SI... ..ENTONCES, de manera sencilla se puede adaptar estas reglas teóricas para el cambio de los procesos de muestreo a las reglas utilizadas como conocimiento heurístico en la base de conocimiento de algún sistema experto que tenga la finalidad de dar como resultado ese cambio de inspecciones a través de un programa.

A continuación se muestran estas reglas expresadas en la sintaxis del cascarón de sistemas expertos I2+.

!las metas son:

!

1. inspeccion / WHAT

1.1 inspeccion2 / WHAT

!

!las reglas son las siguientes:

!

RULE 0

IF desea \ si

AND muestreo \ único

AND tamaño lote >=0

AND riesgo >=0

THEN tiene parámetros

!

RULE 1

IF tiene parámetros

AND dos cinco rechazados \ si

THEN inspección \ rigurosa

AND DISPLAY rigurosa1

!

RULE 2.0

IF tiene parámetros

AND dos cinco rechazados \ no

AND los diez \ si

AND el total de defectos \ si

AND producción estable \ si

AND aprobado \ si

THEN inspección \ reducida

AND DISPLAY reducida1

!

RULE 2.1

IF tiene parámetros

AND dos cinco rechazados \ no

AND los diez \ no

OR el total de defectos \ no

OR producción estable \ no

OR aprobado \ no

THEN inspección \ normal

AND DISPLAY normal1

!

RULE 3.0

IF tiene parámetros

AND inspección \ rigurosa

AND lotes consecutivos \ no

AND diez lotes \ si

THEN inspección2 \ discontinuar

AND DISPLAY discontinuar1

!

RULE 3.1

IF tiene parámetros

AND inspección \ rigurosa

AND lotes consecutivos \ no

AND diez lotes \ no

THEN inspección2 \ rigurosa

AND DISPLAY rigurosa2

AND FORGET inspección

AND FORGET lote

AND FORGET lote aceptado

AND FORGET producción irregular

AND FORGET condiciones

AND FORGET lotes consecutivos

AND FORGET diez lotes

AND FORGET dos cinco rechazados

AND FORGET los diez

AND FORGET el total de defectos

AND FORGET producción estable

AND FORGET aprobado

AND FORGET inspección2

AND CYCLE

!

RULE 4.0

IF tiene parámetros

AND inspección \ rigurosa

AND lotes consecutivos \ si


```
THEN inspección2 \ normal
AND DISPLAY normal2
AND FORGET inspección
AND FORGET lote
AND FORGET lote aceptado
AND FORGET producción irregular
AND FORGET condiciones
AND FORGET lotes consecutivos
AND FORGET diez lotes
AND FORGET dos cinco rechazados
AND FORGET los diez
AND FORGET el total de defectos
AND FORGET producción estable
AND FORGET aprobado
AND FORGET inspección2
AND CYCLE
```

!

RULE 5.0

```
IF tiene parámetros
AND inspección \ reducida
AND lote \ no
AND lote aceptado \ no
AND producción irregular \ no
AND condiciones \ no
THEN inspección2 \ reducida
AND DISPLAY reducida2
AND FORGET inspección
AND FORGET lote
AND FORGET lote aceptado
AND FORGET producción irregular
AND FORGET condiciones
AND FORGET lotes consecutivos
AND FORGET diez lotes
AND FORGET dos cinco rechazados
AND FORGET los diez
AND FORGET el total de defectos
AND FORGET producción estable
AND FORGET aprobado
AND FORGET inspección2
AND CYCLE
```

!

RULE 5.1

```
IF tiene parámetros
AND inspección \ reducida
AND lote \ si
OR lote aceptado \ si
OR producción irregular \ si
```

OR condiciones \ si
THEN inspección2 \ normal
AND DISPLAY normal2
AND FORGET inspección
AND FORGET lote
AND FORGET lote aceptado
AND FORGET producción irregular
AND FORGET condiciones
AND FORGET lotes consecutivos
AND FORGET diez lotes
AND FORGET dos cinco rechazados
AND FORGET los diez
AND FORGET el total de defectos
AND FORGET producción estable
AND FORGET aprobado
AND FORGET inspección2
AND CYCLE

!
!los textos son los siguientes:

!
TEXT desea
Se debe empezar por una inspección normal (utilizando tablas correspondientes a la inspección normal y al tipo de muestreo único) y se utilizara un tipo de muestreo único. ¿Desea empezar la inspección?

!
TEXT muestreo
Existen tres tipos de muestreo único, doble y múltiple, en este caso se utilizara un muestreo único.

!
TEXT tamaño lote
Introduzca el numero correspondiente al tamaño del lote.

!
TEXT riesgo
Introduzca el AQL o riesgo del productor.

!
TEXT producción estable
¿La producción es estable?

!
TEXT aprobado
¿Se aprueba por persona autorizada?

!
TEXT lote
¿Lote rechazado?

!
TEXT condiciones
¿Lo aconsejan otras condiciones?

!

```
TEXT dos cinco rechazados
¿Dos de cinco lotes consecutivos rechazados?
!
TEXT los diez
¿Los diez lotes precedentes se aceptaron?
!
TEXT el total de defectos
¿El total de defectos (rechazos) fue menor que el número limite (tabla w)?
!
DISPLAY normal2
Regrese a la inspección normal
!
DISPLAY rigurosa1
Cambie a la inspección rigurosa
!
DISPLAY rigurosa2
Permanezca en inspección rigurosa
!
DISPLAY discontinuar1
Inspección discontinua
!
DISPLAY normal1
Continué en la inspección normal
!
DISPLAY reducida2
Siga en la inspección reducida
!
DISPLAY reducida1
Cambie a la inspección reducida
!
END
```

Sin Embargo, en este ejemplo se muestran los procesos de aceptación o rechazo de lotes, así como los procesos de cambio de régimen de inspección.

Con lo anterior en mente, las premisas como:

```
AND dos cinco rechazados \si
AND los diez \si
AND el total de defectos \si
```

Se muestran en este ejemplo como información que deberá proporcionar el usuario.

En la practica el sistema experto puede estructurarse de tal manera que los únicos datos que requiere del usuario son:

- **Tamaño del lote.**
- **Nivel de AQL.**

- **El número de rechazados de un lote que se este inspeccionando.**

En este último caso, el sistema puede almacenar los resultados históricos de las inspecciones, consultadas y llegar inmediatamente a conclusiones como:

AND dos cinco rechazados \si

AND los diez \si

AND el total de defectos \si

Este es el caso del sistema que se desarrolló en esta tesis.

CAPITULO 3

SELECCIÓN DE LOS RECURSOS

En este capítulo se describe el alcance de las herramientas que estuvieron a nuestra disposición para poder desarrollar el software del cual trata esta tesis o mejor dicho el sistema experto muestreo para aceptación en programas de control de calidad. Además se dan a conocer las características esenciales de todas estas herramientas y mencionar cuales se descartaron y cuales se aceptaron en el desarrollo de este sistema.

3.1 NECESIDADES.

Se tiene la necesidad de estructurar y programar un sistema experto con recursos informáticos y tecnológicos al alcance de cualquier persona que cuente con una computadora personal de escritorio y sin la necesidad de pagar licencias por el uso de software, en la que el usuario pueda tener acceso a dicho sistema experto a través de una interfase gráfica de usuario, amigable, de fácil manejo con el uso de menús, botones y ventanas con colores agradables a la vista y que todos los elementos de interacción se operen mediante el ratón y el teclado.

Además no es necesario contar con una estructura de cliente/servidor, debido a que la base de datos puede ser local. Así el shell, la base de datos y el lenguaje de programación se deben encontrar instalados en una sola PC.

Para la base de datos no es necesario utilizar las características de una base de datos relacional debido a que, solo se emplean tablas independientes, es decir cada tabla creada no se relaciona con la otra por medio de llaves foráneas (no tienen campos comunes), ni cuentan con llaves primarias.

3.2 ANÁLISIS DE RECURSOS.

A continuación se hace un análisis de todos los recursos seleccionados o candidatos a ser usados en el desarrollo del sistema, describiendo su entorno de trabajo.

3.2.1 BASES DE DATOS.

Las bases de datos son sistemas electrónicos de clasificación de archivos y almacenan información.

Un gestor de base de datos es un programa que permite introducir y almacenar datos, ordenarlos y manipularlos. Organizarlos de manera significativa para que se pueda obtener información no visible como totales, tendencias o relaciones de otro tipo. Debe permitir en principio: introducir datos, almacenar datos, recuperar datos y trabajar con ellos.

3.2.1.1 EJEMPLOS DE BASES DE DATOS.

Excel:

Excel 2000 es una aplicación del tipo *hoja de cálculo*, integrada en el entorno *Windows* y desarrollada por *Microsoft*, en la cual se combinan las capacidades de una hoja de cálculo normal, base de datos, programa de gráficos bidimensionales y tridimensionales, lenguaje propio de programación y generación de macros; todo dentro de la misma aplicación. Una *hoja de cálculo* es una especie de **tabla** cuyas *casillas* o *celdas* pueden contener texto, valores numéricos, fechas y datos horarios, fórmulas o funciones matemáticas.

En *Excel* se identifica el concepto de base de datos con el de *lista* (o *tabla*). Por lo tanto una base de datos en *Excel* es un conjunto de filas en las cuales una columna tiene siempre el mismo tipo de información.

La definición que da *Excel* de *lista* es:

“Serie rotulada de filas de hojas de cálculo que contiene grupos similares de datos, como una base de datos de facturas o un conjunto de nombres y números telefónicos de clientes. Una lista puede usarse como una base de datos, donde las filas son registros y las columnas son campos.”

Durante la generación de la hoja de cálculo, no es necesario definir ninguna propiedad específica para realizar operaciones de base de datos como introducir, borrar, buscar, ordenar o efectuar el subtotal de los datos, el programa reconoce la lista como una base de datos.

Para crear una base de datos o una lista en *Excel* se debe introducir una primera fila en la que se describe el contenido de cada columna. A continuación se introducen en las filas siguientes cada uno de los datos. Estas filas constituyen los registros de la lista. Como se muestra en la Tabla 3.2.1.1.1

	A	B	C	D	E
1	Camet	Nombre	Apellido 1	Apellido 2	Curso
2	90125	Pablo	Garcés	Moral	1
3	91487	Pedro Jos	García	García	2
4	94564	Jaime	Inchausti	Zatrás	3
5	94287	Igor	Maturana	Calvo	3

Tabla 3.2.1.1.1

Access:

Access es una aplicación de base de datos. Una aplicación de base de datos que almacena y gestiona información se denomina un sistema de gestión de bases de datos SGBD ó DBMS (*Data Base Management System*). La estructura básica de esta información son tablas. Para una mejor gestión de la información y para reducir el número de datos, las tablas se relacionan entre ellas. Debido a este carácter relacional, *Access* es un sistema de gestión de bases de datos **relacional** SGBDR ó RDBMS (*Relational Data Base Management System*).

3.2.2 LENGUAJES DE PROGRAMACIÓN.

Programar es almacenar en memoria la información sobre la tarea que las computadoras van a ejecutar. Las primeras se usaban como calculadoras simples; que les indicaban los pasos de cálculo, uno por uno.

Los lenguajes más primitivos fueron los lenguajes de máquina. Esto, ya que el hardware se desarrolló antes del software, y además cualquier software finalmente tiene que expresarse en el lenguaje que maneja el hardware.

La programación es sumamente tediosa, pues el programador tenía que "bajarse" al nivel de la máquina y decirle, paso a paso, cada punto de la tarea que tenía que realizar. Además, debía expresarlo en forma numérica; y por supuesto, este proceso era propenso a errores, con lo que la productividad del programador era muy limitada.

El primer gran avance que se dio, fue la abstracción dada por el Lenguaje Ensamblador, y con él, el nacimiento de las primeras herramientas automáticas para generar el código máquina. Sin embargo, para el programador es fácil perderse y cometer errores de lógica, pues debe bajar al nivel de la forma en que trabaja el CPU, y entender bien todo lo que sucede dentro de él.

El Lenguaje de Alto Nivel, nació con el primer compilador de FORTRAN (FORMula TRANslation), que, inició como un simple esfuerzo de traducir un lenguaje de fórmulas, al lenguaje ensamblador y por consiguiente al lenguaje de máquina.

Algunas de las características que los lenguajes de programación deben tener son:

- **Robusto.** El lenguaje debe manejar la memoria de la computadora por el usuario. El usuario no se tiene que preocupar por apuntadores, memoria que no se esté utilizando, etc. el programa debe realizar todo esto sin necesidad de que el usuario se lo indique.
- **Seguro.** El lenguaje debe tener ciertas políticas que eviten que se puedan codificar virus.
- **Portable.** Un programa compilado de este lenguaje puede ser utilizado por cualquier computadora que tenga implementado su interprete .
- **Multithreaded.** El programa debe ser Multithreaded. Un thread o un thread de control es una serie de instrucciones (executable steps). Un thread de control es un programa en ejecución. Cada thread posee todo lo necesario para su ejecución: una pila, código y segmentos de datos. Dado que un thread se ha definido como una secuencia de instrucciones a ser ejecutadas, un proceso es un thread. Cualquier tarea que se completa usando más de uno de estos threads puede ser considerada **multithreaded**.

3.2.2.1 EJEMPLOS DE LENGUAJES DE PROGRAMACIÓN.

Visual Basic:

Visual Basic es un lenguaje de programación visual en donde gran número de tareas se realizan sin escribir código, simplemente con operaciones graficas realizadas con el ratón sobre la pantalla.

Es también un programa basado en objetos, aunque no orientado a objetos como **c++** o **java**, debido a que utiliza objetos con propiedades y métodos pero carece de los mecanismos de **herencia y polimorfismo** propios de los verdaderos lenguajes orientados a objetos como **java** y **c++**.

Visual Basic es un lenguaje **orientado a eventos** como los programas típicos de **Windows** tales como **Netscape, Word, Excel y PowerPoint**. Estos programas pasan la mayor parte del tiempo esperando las acciones del usuario(**eventos**) y respondiendo a ellas.

Visual Basic está orientado a la realización de programas para **Windows**, pudiendo incorporar todos los elementos de este entorno informático: ventanas, botones, cajas de dialogo y de texto, botones de opción y de selección, barras de desplazamiento, gráficos, menús ,etc. Prácticamente todos los elementos de interacción con el usuario de los que dispone **Windows 95/98/NT** pueden ser programados en **Visual Basic**.

Dispone de un **modo de diseño** y un **modo de ejecución**. En **modo de diseño** el usuario construye interactivamente la aplicación, colocando controles en el **formulario**, definiendo sus propiedades, y desarrollando funciones para gestionar los eventos. La aplicación se prueba en **modo de ejecución**. En este caso el usuario actúa sobre el programa y prueba como responde el programa. Hay algunas propiedades de los controles que deben establecerse en **modo de diseño**, pero muchas otras pueden cambiarse en tiempo de ejecución desde el programa escrito en **Visual Basic**. También hay propiedades que solo pueden establecerse en **modo de ejecución** y no son visibles en **modo de diseño**.

Cada uno de los elementos gráficos que forman parte de una aplicación típica de **Windows 95/98/NT** es un tipo de control como los botones, las cajas de dialogo y de texto, los gráficos, los menús y otros. En **Visual Basic** el **formulario** (form) es una ventana y es considerado como un contenedor de controles.

Los formularios y los distintos tipos de controles son entidades genéricas de las que puede haber varios ejemplares concretos en cada programa. En la programación **orientada a objetos** se llama **clase** a estas entidades genéricas, mientras que se llama **objeto** a cada ejemplar de una clase determinada.

Cada vez que se produce un **evento** sobre un determinado tipo de control, **Visual Basic** arranca una determinada **función** o **procedimiento** que realiza la acción programada por el usuario para ese evento concreto.

Los **métodos** son funciones que también son llamadas desde programa, pero a diferencia de los procedimientos no son programadas por el usuario si no que vienen ya pre-programadas con el lenguaje.

Se requiere 50 MB en disco duro procesador Pentium y 16 MB en Ram.

Visual C++

Para desarrollar programas, **Windows** provee una librería de rutinas y funciones (SDK - Kit de desarrollo de software) que permiten gestionar componentes como menús, diálogos, ventanas, etc.

Visual C++ es un entorno integrado de desarrollo que permite la programación **orientada a objetos (POO)** conjuntamente con el sistema de desarrollo **SDK** (también denominado API) de Windows. Al ser un entorno integrado **Visual C++** incluye, entre otras, las siguientes herramientas de desarrollo:

- Editor de texto
- Compilador/Enlazador
- Depurador
- Visor de datos y dependencias (Browser)

El **SDK** de **Windows** es un complejo conjunto de funciones que añade además numerosas definiciones de tipos de datos nuevos para cualquier programador de **C/C++ para DOS**. Para solucionar este problema, **Visual C++** incluye la librería de clases **MFC** (Microsoft Foundation Classes) que permite crear y gestionar de manera intuitiva componentes típicos de Windows. Esto es, la **MFC** es una implementación que utiliza el **API** encapsulando todas las estructuras y llamadas a funciones en objetos fáciles de utilizar. Basándose en la potencia de la **MFC**, **Visual C++** se convierte en un generador de programas **C++ para Windows**.

“**C**” es un lenguaje de alto nivel, basado en funciones, que permite desarrollos estructurados. Entre otras muchas características contempla la definición de estructuras de datos, recursividad o direcciones a datos o código (punteros).

“**C ++**”, por su parte, es un superconjunto de “**C**”, al que recubre con una capa de soporte a la **POO**. Permite por tanto la definición, creación y manipulación de objetos.

La **POO** se basa en la utilización de objetos. El objetivo de la **POO** es la meta de cualquier modelo de programación estructurada convencional: “imponer” una serie de normas de desarrollo que aseguren y faciliten la mantenibilidad y reusabilidad del código.

Los mecanismos básicos de la **POO** son: objetos, mensajes, métodos y clases.

Las principales características de la **POO** son: abstracción, encapsulamiento, herencia y polimorfismo.

Se requiere procesador Pentium® 90MHz (mínimo) 24MB Ram (32MB Ram recomendado) y 290Mb en disco duro.

3.2.3 SISTEMAS OPERATIVOS.

Con el software, una computadora puede almacenar, procesar y recuperar información, encontrar errores de ortografía e intervenir en muchas otras valiosas actividades. **El software**

para computadoras puede clasificarse en dos clases: los programas de sistema, que controlan la operación de la computadora en sí y los programas de aplicación, los cuales resuelven problemas para sus usuarios. El programa fundamental de todos los programas de sistema, es el **Sistema Operativo**, que controla todos los recursos de la computadora y proporciona la base sobre la cual pueden escribirse los programas de aplicación.

Un **Sistema Operativo** es un programa que actúa como intermediario entre el usuario y el hardware de un computador y su propósito es proporcionar un entorno en el cual el usuario pueda ejecutar programas. El objetivo principal de un **Sistema Operativo** es, entonces, lograr que el Sistema de computación se use de manera cómoda, y el objetivo secundario es que el hardware del computador se emplee de manera eficiente.

3.2.3.1 EJEMPLOS DE SISTEMAS OPERATIVOS.

WINDOWS 98:

Windows 98 es un sistema operativo, es decir el encargado de controlar y dirigir a la computadora traduciendo las instrucciones a un lenguaje que el *hardware* (partes físicas de la computadora) puede comprender. La diferencia con el sistema operativo tradicional está en la manera intuitiva y mediante dibujos o *iconos* de comunicarse con la PC en vez de teclear comandos.

En *Windows 98* Los nombres de ficheros y directorios en pueden contener hasta 255 caracteres, blancos incluidos.

Windows 98 permite realizar una misma acción copiar un fichero, arrancar una aplicación, etc. de varias formas. El usuario puede decidir cuál de ellas le resulta más cómoda.

A continuación se enumeran los requisitos mínimos de hardware para **Windows 98**. También se enumeran los componentes opcionales que puede instalar.

En la lista siguiente se describen los requisitos mínimos de hardware para Windows 98:

- Un equipo personal con un 486DX 66 megahercio (MHz) o un procesador más rápido (unidad central recomendada de procesamiento Pentium).
- 16 megabytes de memoria (24 MB recomendado) (MB).
- Una actualización típica desde Windows 95 requiere aproximadamente 195 MB de espacio libre en el disco duro pero el espacio en el disco duro puede ir de entre MB 120 y 295 MB dependiendo de su configuración de equipo y dependiendo de las opciones a las que elige que instalan.
- Una instalación completa de Windows 98 en una unidad FAT16 requiere 225 MB de espacio libre en el disco duro pero puede ir de entre 165 MB y 355 MB dependiendo de su configuración de equipo y dependiendo de aquellas opciones a las que elige que instalan.
- Una instalación completa de Windows 98 en una unidad FAT32 requiere 175 MB de espacio libre en el disco duro pero puede ir de entre 140 MB y 255 MB dependiendo de su configuración de equipo y dependiendo de las opciones a las que elige que instalan.
- Una unidad de disco 3.5-inch de densidad alta.

- VGA o mayor resolución (color de 16 bits o 24 bits que se recomendó por SVGA).

El acceso a MSN, a The Microsoft Network, a Mensajería de Windows o a Internet requiere unos 14,4 bits por segundo módem (bps) 28,8 o recomendado más rápidamente.

Componentes opcionales:

- Adaptador de red
- Adaptador DVD-ROM de unidad y de decodificador
- Tarjeta y altavoces de audio para sonido
- Escáner o cámara digital
- Segundo adaptador de vídeo de monitor y de segundo
- Bus IEEE 1394 (Firewire)
- Bus universal Bus Serie (USB) y hardware HID
- Tarjeta ATI gráfica All-in-Wonder

Windows 98 es un sistema operativo de multiprogramación (o Sistema Operativo de multitarea). Estos sistemas se distinguen por sus habilidades para poder soportar la ejecución de dos o más trabajos activos (que se están ejecutando) al mismo tiempo. Esto trae como resultado que la Unidad Central de Procesamiento (CPU) siempre tenga alguna tarea que ejecutar, aprovechando al máximo su utilización.

3.2.4 SHELL PARA SISTEMAS EXPERTOS.

Históricamente, los primeros Sistemas Basados en Conocimiento fueron desarrollados utilizando lenguajes de programación como el LISP y el PROLOG. A medida que el desarrollo de Sistemas Basados en Conocimiento iba aumentando en cantidad y complejidad, la comunidad científica comenzó a buscar formas de desarrollar los sistemas en menor tiempo y con menor esfuerzo.

Esto dio lugar al apareamiento, de sistemas vacíos como el EMYCIN, a los que denominó shells, ya que ofrecen toda la arquitectura de un Sistema Basado en Conocimiento a los que hay que incorporar la base de conocimientos.

3.2.4.1 EJEMPLOS DE SHELL.

CLIPS:

Clips es una herramienta de software para la creación de sistemas expertos. Un sistema experto es un programa de computadora pensado para realizar las tareas que normalmente realiza un ser humano considerado "experto" en algún área. La finalidad del sistema experto es realizar la tarea del experto humano reproduciendo el mismo modelo que éste utiliza para la resolución de problemas pertenecientes a su ámbito de saber. De este modo como se menciona en el capítulo uno, se pueden distinguir las siguientes partes:

- Los datos, o información base sobre el campo en el que se desarrollarán las actividades del experto.

- La base de conocimiento, que recoge las formas de utilizar la información citada en el punto anterior.
- El motor de inferencias, que modela el proceso de razonamiento mediante el cual el experto humano realiza su tarea.

Clips es una herramienta para construir sistemas expertos basados en reglas. Las reglas son la forma de representar los conocimientos del experto constan de una parte condicional que, si se cumple, tiene como consecuencia la ejecución de una acción asociada.

Cada vez que se aplica una de las reglas, los datos guardados en el sistema pueden variar, dando lugar a que se activen otras reglas que en la situación anterior no lo estaban. La resolución del problema planteado pasa por la aplicación sucesiva de reglas hasta que se llega a la condición de finalización. A este proceso se denomina **conducido por los datos**, ya que las acciones solo se producen cuando hay datos que las justifiquen.

3.3 SELECCIÓN DE LOS RECURSOS A UTILIZAR.

En este tema se darán algunas consideraciones del porqué de la utilización de estas herramientas así como algunos aspectos de sus entornos de desarrollo.

3.3.1 WINDOWS 98 COMO SISTEMA OPERATIVO.

Se eligió **Windows** porque proporciona ventajas significativas tanto a los usuarios como a los desarrolladores sobre al convencional entorno de **MS-DOS**.

Además de que **Windows** es una interfase grafica de usuario (**GUI: graphical User Interfase**). Es un sistema operativo monousuario y multitarea. Puede ejecutar aplicaciones para **Windows** y aplicaciones para **DOS**.

Windows facilita la comunicación entre el usuario y la computadora, a través de ventanas las cuales le proporcionan al usuario información y controles por medio de los cuales el usuario le envía información a la computadora.

La información que las ventanas de **Windows** muestran al usuario puede ser de texto o grafica, los controles vienen incluidos en las ventanas y le permiten al usuario realizar ciertas operaciones como por ejemplo seleccionar de una lista de valores, aceptar al seleccionar un botón o introducir texto en una caja de texto.

El sistema operativo **Windows** es de fácil acceso y de fácil manejo, a través del ratón para el usuario hacia el cual va dirigido el sistema experto, además de que es una plataforma en la que se puede ejecutar **Visual Basic** como lenguaje de desarrollo al igual que **Clips**, así como las aplicaciones de **Access** y de **Excel** para el manejo de bases de datos

3.3.2 VISUAL BASIC 6.0 COMO ENTORNO DE DESARROLLO.

Se eligió Visual Basic 6.0 como lenguaje de programación o entrono de desarrollo de sistema experto entre otras razones porque ya se tenía conocimiento de su funcionamiento y estructura de programación ahorrando tiempo en el aprendizaje de otro programa.

Además la interfase de Visual Basic contiene todos los elementos con que la interfase de Windows trabaja por lo que consta de un ambiente gráfico amigable y cumple con las expectativas de la programación orientada a objetos aunque no incluye la herencia y el polimorfismo.

Por lo que respecta a la interacción con las bases de datos Visual Basic 6.0 es compatible con Access y Excel. Visual Basic se conecta a la base de datos a través de un DataControl. Un Datacontrol es un tipo de control propio de Visual Basic pensado específicamente para el acceso a la base de datos como se observa en la figura 3.3.2.1.

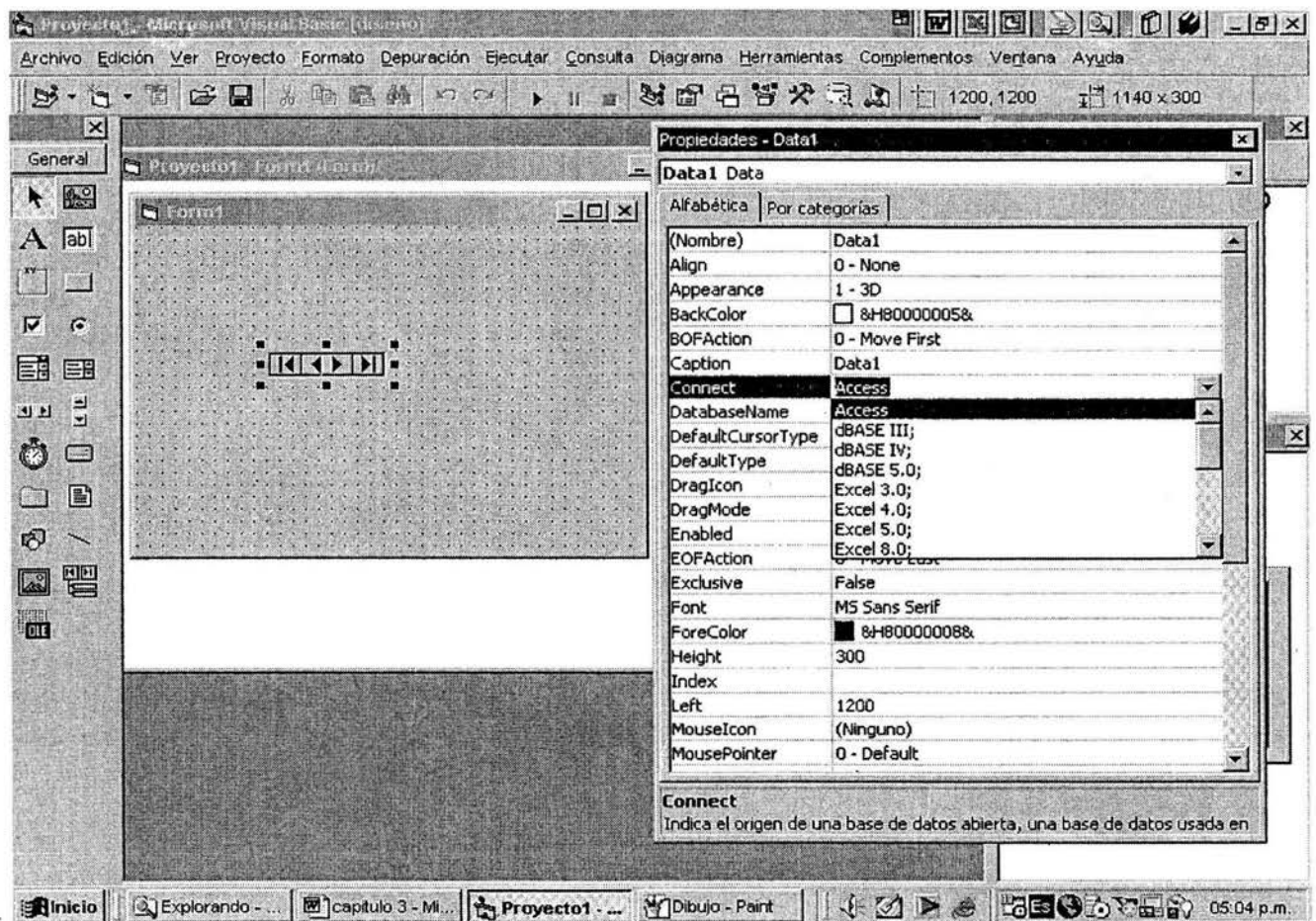


Figura 3.3.2.1

3.3.3 ACCESS 2000 Y EXCEL 2000 COMO BASES DE DATOS.

El sistema experto es un software pensado para que se ejecute en una PC de escritorio, y es una ventaja que cualquier computadora personal cuente con la suite de productos de Microsoft entre ellos Access y Excel. Estas son aplicaciones que cubren las necesidades de bases de datos del sistema experto propuesto.

Tanto Access como Excel fueron herramientas requeridas para la construcción del sistema experto.

Se utilizó Excel para crear bases de datos propias del estándar militar que se empleó para desarrollar esta tesis, dado que las bases de datos del estándar son grandes, es decir cuentan con demasiados registros y campos.

Dada la habilidad que tiene Excel de manejar renglones y columnas es más fácil manipular gran número de datos en esta hoja de cálculo, además que estas bases de datos no son relacionales por lo que se usan únicamente para sacar información de ellas, debido a esto Excel ajusto perfectamente en su construcción.

Access se utilizó para construir también bases de datos no relacionales pero la estructura de estas bases es de menor tamaño. Dado que el sistema experto necesita llevar un control de la información que el usuario le proporciona Access es apropiado para guardar esta información directamente en la ejecutando el sistema.

3.3.4 CLIPS COMO MÁQUINA DE INFERENCIAS.

La selección del shell se centró en un software llamado Clips, el cual implementa métodos de inferencia con encadenamiento hacia delante, y es compatible con Visual Basic a través de una librería llamada Clips ActiveX OCX la cual se encarga de la comunicación de estas dos herramientas. Al instalar Clips ActiveX OCX, se crea en Visual Basic un nuevo control llamado Clips el cual cuenta con todos sus comandos ya preestablecidos en su entorno de desarrollo normal. Clips es visto entonces como un nuevo objeto en Visual Basic y éste llama a sus comandos de la misma forma que cualquier otro objeto propio de éste lenguaje llama a sus métodos.

La herramienta Clips incluye un entorno de desarrollo para programar, sistemas de forma completa. En la ventana de comandos que ilustra la figura 3.3.3.1 aparece el prompt:
CLIPS>

En este prompt se pueden introducir comandos directamente en lo que se denomina alto nivel, como por ejemplo el comando assert para introducir hechos en el sistema, por ejemplo:
CLIPS> (assert (hecho1)(hecho2)) así para eliminar hechos del sistema se utiliza el comando retract seguido por el número de índice que tenga el hecho.
CLIPS> (retract 0).

En su entorno de desarrollo clips consta de tres archivos:

- El ejecutable (clipswin.exe) es el que controla el interfase de comandos y el mecanismo de inferencia
- El archivo de ayuda (clips6.hpl) contiene referencias a los comandos de Clips
- El editor de texto (Clipsedit.exe) sirve para introducir comandos, pudiendo guardarlos en archivos para usarlos posteriormente.

Ejecutando el archivo “clipswin.exe” aparece la ventana de comandos, con el prompt CLIPS>. Desde este prompt ya se pueden insertar comandos, aunque tiene la desventaja de no ofrecer la posibilidad de copiar y pegar ni de mantener un histórico de comandos. Por eso lo más cómodo es utilizar el editor de texto (ctrl.+E) para guardar los comandos en un archivo y cargarlo después.

3.3.4.1 CLIPS ACTIVEX OCX.

El Clips ActiveX Control es una implementación de la completa ingeniería de Clips como un componente OCX.

Todas las características de Clips están habilitadas, incluyendo la extensión orientada a objetos (Cool) y además construye la compilación (ConstructsToC). Se documentan los detalles del ambiente de programación de Clips en los manuales siguientes. Éstos están disponibles en el sitio Web de Clips:

<http://www.CLIPS Basic Programming Guide>

<http://www.CLIPS Advanced Programrnrning Guide>

<http://www.CLIPS User's Guide>

<http://www.CLIPS Interfaces Guide>

El control OCX no es una interfaz para Windows para el interactivo Clips interpreter/workspace (i.e. no duplica la funcionalidad de clipswin.exe). Para el desarrollo interactivo de Clips, es necesario usar todavía el Clips Windows ejecutable. El OCX le permite empotrar el poder del software de Clips fácilmente en otros programas.

Pueden representarse las estructuras de Clips como objetos de COM discretos (los hechos, las instancias, etc.) haciendo muy fácil de manipularlos en cualquier OLE/OCX aplicación habilitada o ambiente de desarrollo.

La versión Web le permite construir páginas de HTML que pueden usar el OCX de VBScript en el navegador de Internet Explorer. Las direcciones de Clips (es decir: pointers) son manipuladas en el lenguaje del host (por ejemplo: Visual Basic) como los valores de Long Integer así le hacen a Visual Basic acceder fácilmente a algunos de los comandos de Clips de muy bajo nivel. Las rutas de entrada y salida de Clips pueden tratarse como un “set de resultados de tipo string” (result set) o un largo arreglo de tipo cadena.

Éstos pueden ser navegados fácilmente con los comandos FindXXX y MoveXXX, de tal manera que es relativamente sencillo pasar una gran cantidad de datos de Clips a los programas de aplicación lo cual hace relativamente fácil analizar y recuperar estos datos.

Los recursos en línea incluyen:

<http://www.CLIPS Homepage>

<http://www.CLIPS OCX homepage>

<http://www.Searchable CLIPSMailing list archive>

<http://www.CLIPS Developer Forum>

CAPITULO 4

ENTORNO DE DESARROLLO DEL SISTEMA EXPERTO

En este capítulo se mostrarán algunas estructuras de programación en las diferentes herramientas que se ocuparon en el desarrollo de éste sistema, además se trata de ilustrar en diagramas de acuerdo a la ingeniería de sistemas los diferentes subsistemas que componen a este sistema y los pasos ordenados que sigue el sistema como un conjunto de procesos funcionales.

4.1 ELEMENTOS DE PROGRAMACIÓN DE CLIPS.

Se utilizaron algunas sentencias o comandos de Clips para la realización del sistema como las siguiente:

La instrucción **assert** que se utiliza para introducir hechos en la base de hechos de uno en uno o varios hechos en la misma instrucción.

Un ejemplo sería

```
CLIPS> (assert (color tomate rojo))  
CLIPS> (assert (hecho-1) (hecho-2))
```

Para introducir hechos por primera vez hay que reiniciar el sistema con el comando **reset**. Este comando limpia la base de hechos, e introduce el hecho (initial-fact)

La instrucción **defrule** sirve para definir reglas como la siguiente:

```
(defrule pasar  
      (color luz-semaforo verde)  
=>  
      (assert (se-puede-cruzar si)))
```

- La parte de antes del símbolo “=>” se denomina “parte izquierda de la regla” (o antecedente) y está compuesta por patrones que son cada una de las condiciones que se deben cumplir en la base de hechos para que la regla se active. En el ejemplo anterior solo hay un patrón.
- La parte que aparece con posterioridad al símbolo “=>” se denomina “parte derecha de la regla” (o consecuente) y está formada por cada una de las acciones que tendrán lugar cuando todos los patrones de la parte izquierda correspondan con los hechos de la base de hechos.

La prioridad de cada regla se estableció mediante la propiedad **salience** que se puede asociar a una regla cuando se define. Este valor varia entre -10000 y 10000, valiendo cero por defecto.

Por ejemplo: las siguientes reglas todas son candidatas a dispararse debido a que existen los hechos en la base de hechos que corresponden con su antecedente, pero según el salience más

alto se determina cual se disparará primero que las demás, en este caso será la regla “esperar” debido a que su salience es él más alto de las dos, siguiéndole la regla “pasar”.

```
(defrule pasar
  (declare (salience 1))
  (color luz-semaforo verde)
=>
  (assert (se-puede-cruzar si)))

(defrule esperar
  (declare (salience 4))
  (color luz-semaforo amarilla)
=>
  (assert ( prepararse-para-cruzar si)))
```

La construcción **defglobal** define variables que son visibles en todo el entorno de Clips, es decir permite definir variables globales que pueden ser accesadas desde cualquier punto en Clips y retienen su valor independiente de otras construcciones.

Su sintaxis es la siguiente:

```
CLIPS>(defglobal ? *variableA*=12)
CLIPS>(defglobal ? *variableA*=12 ? *variableC*=2 ? *variableD*=3)
```

Para la asignación de un valor a una variable en el consecuente de una regla se utiliza la sentencia **bind**: en el antecedente de una regla se le asigna un valor a una variable mediante la correspondencia del patrón con un hecho de la base de hechos, mientras que en el consecuente de la regla se hace mediante la función bind.

Por ejemplo:

```
(defrule cumpleaños
  ?índice1<- (persona José Cano edad 19)
=>
  (bind ?índice2 (persona José Cano edad 20))
```

Para eliminar hechos en una regla hay que asignarle primero a una variable el índice del hecho en el antecedente de la regla.

El índice del hecho se especifica en Clips utilizando los símbolos “<- “. Después aplicamos el comando **retract** seguido de la variable que almacena el índice del hecho. Esto borrara de la base de hechos el hecho deseado.

Un ejemplo sería:

(defrule desincribir-jose cano

```
?huesped<- (huesped José Cano)
```

=>

```
(retract ?huesped))
```

Para la salida de Clips se emplea la sentencia **printout** y se indica con una **t** si se desea que esta sea en pantalla.

De esta forma se imprimen mensajes en Clips.

```
CLIPS>(printout t "HOLA" crlf)
```

El comando **reset** no borra las reglas que hay activas en la agenda, actúa sobre los hechos. El comando **clear** borra tanto hechos como reglas.

Una vez que se tiene completa la base de hechos y las reglas del sistema, el sistema se ejecuta mediante la sentencia **run**.

```
CLIPS>(run)
```

Con esta sentencia, la ejecución se realiza de forma interrumpida, hasta que no haya más reglas activas en la agenda, pero se puede indicar que no se ejecute el programa hasta el final, sino un número fijo de pasos o reglas poniendo el número deseado en la sentencia **run**.

```
CLIPS>(run 3)
```

4.2 COMUNICACIÓN INTERNA DEL SHELL CLIPS.

La impresión de mensajes en pantalla hace posible una salida visible para el usuario en el entorno de trabajo de Clips, con lo que se puede comprobar el funcionamiento de los programas. Usando la sentencia **printout** y utilizando la letra **t** se indica que la salida sea en pantalla. Por ejemplo:

```
;auto
(defrule uno_es
  (declare (salience 10))
=>
  (printout t " el motor prende si / no:")
  (assert (prende (read)))
)
```

El comando **read** establece una entrada por medio del teclado para que la regla pueda obtener información que el usuario introduzca desde el teclado. El comando **read** lee solo un símbolo y para leer más de una palabra se debe encerrar la frase entera entre comillas. Esto es ilustrado en la figura 4.2.1, que muestra la ventana de trabajo de Clips llamada Dialog y el archivo `auto.clp` abierto en bloc de notas que contiene la base de conocimiento del sistema experto llamado `auto`.

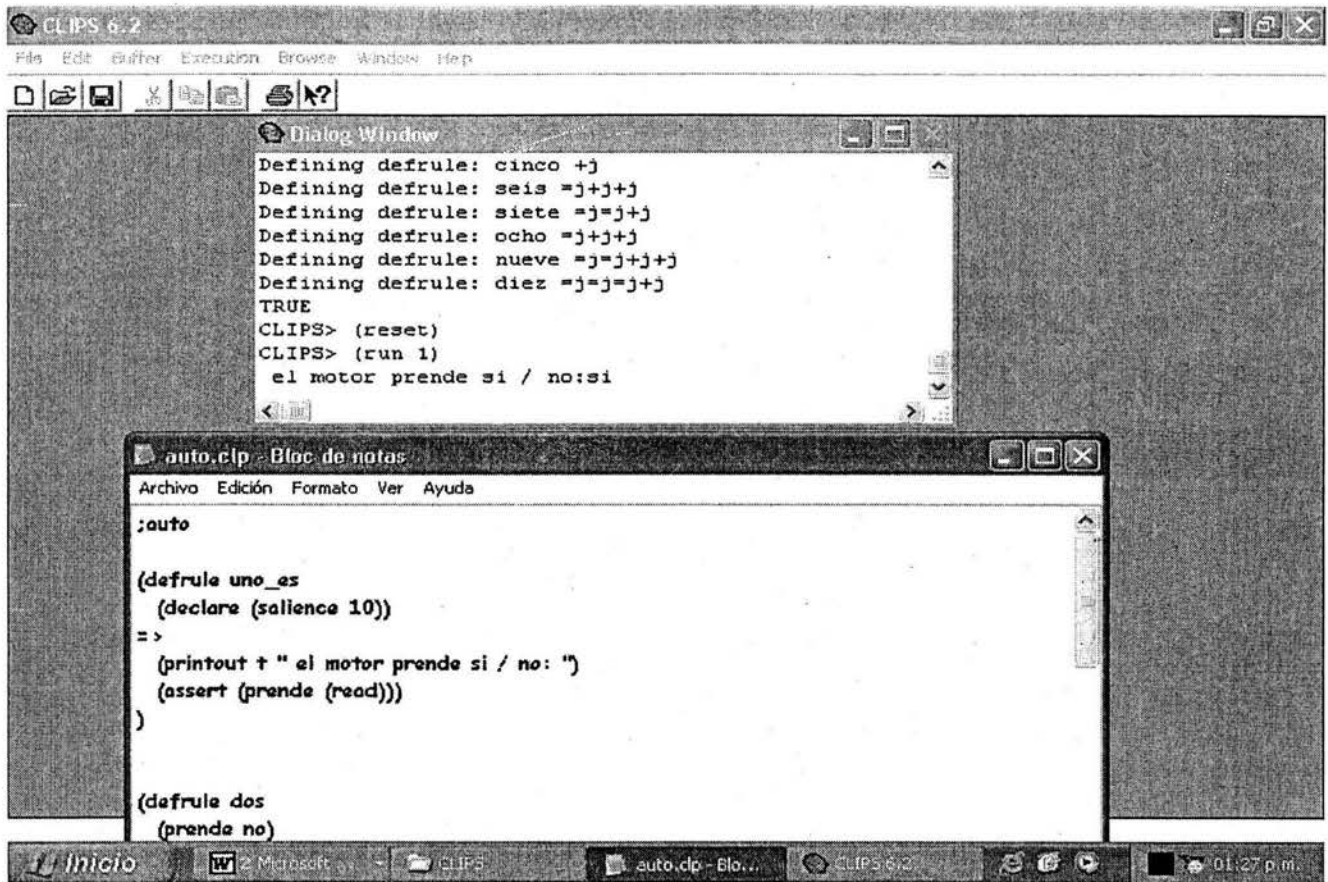


Figura 4.2.1

4.3 BASE DE CONOCIMIENTO DEL SISTEMA EXPERTO MUESTREO PARA ACEPTACIÓN.

A continuación se muestra el código de la base de conocimiento del sistema experto muestreo de aceptación en programas de control de calidad en Clips, donde se aprecian algunos elementos de la sintaxis de Clips que se utilizaron para su realización.

```
(defglobal
    ?*pregunta* = ""           ; se definen variables globales de tipo string y se inicializan con una
                               ;cadena vacía
)
(defglobal
    ?*pregunta1* = ""
)
(defglobal
    ?*solucion* = ""
)
(defglobal
    ?*solucion1* = ""
)
(defglobal
    ?*solucion2* = ""
)
(defglobal
    ?*solucion3* = ""
)
(defglobal
```

```

?*solucion4* = "")
(defglobal
?*solucion5* = "")

(defglobal ?*yaresultado* = 0)

(defrule pedir
;se define una regla.
=>
(bind ?*pregunta* "PROPORCIONE EL TAMAÑO DEL LOTE" ) ;a la variable global pregunta
) ;se le modifica su valor con una
; cadena de caracteres.

(defrule pediraql
; premisa que se debe de cumplir o debe de haber un hecho
(letra ?tama) ; igual en la base de hechos de Clips para que la regla se pueda
=> ;disparar.
(bind ?*solucion* la letra codificada es ?tama)
(bind ?*pregunta* "PROPORCIONE EL RIESGO DEL PRODUCTOR".)
(assert (solo_p)) ;introducción de un hecho a la base de hechos de Clips.
)

(defrule teneriesgo
?r<-(solo_p) ;asignación de un hecho a una variable si el hecho existe.
?a1<-(tiene_aql ?aql2)
?a2<-(tiene_mm ?mm2)
?a3<-(tiene_ac ?ac2)
?a4<-(tiene_re ?renor)
=>
(bind ?*solucion* Tiene una muestra aleatoria de ?mm2 piezas a inspeccionar)
(bind ?*solucion1* "El riesgo del productor es" ?aql2)
(bind ?*solucion2* Se aceptará el lote con ?ac2 se rechazará el lote con ?renor)
(assert (tiene_todo))
(assert (parametro modi))
(retract ?r) ;se borran hechos de la base de hechos de Clips.
(retract ?a1)
(retract ?a2)
(retract ?a3)
)

(defrule todo
(declare (salience 12)) ;se declara la prioridad de la regla.
(tiene_todo)
?E<-(parametro modi)
=>
(bind ?*pregunta* INDIQUE EL NÚMERO DE DEFECTUOSOS O DEFECTOS EN ESA
MUESTRA)
(assert (tiene numdef))

```

```
(retract ?E)
)
```

```
(defrule conta
(declare (salience 10))
(tiene numdef) ;hechos que se deben cumplir para que la regla se dispare.
(tiene_todo)
(tomo ?re1)
(tiene_re ?renor)
(test (= ?*yaresultado* 5)) ;condición que se cumple si la variable "yaresultado" es igual a 5.
=>
(bind ?*yaresultado* 0)
)
```

```
(defrule acep
(declare (salience 9))
(tiene_todo)
(tiene numdef)
?F<-(tomo ?re1)
(tiene_re ?renor)
(test (< ?re1 ?renor)) ;condición que se cumple si la variable re1 contiene un valor menor a la
=> ;variable renor.
(bind ?*solucion3* 0)
(bind ?*yaresultado* (+ ?*yaresultado* 1)) ;se incrementa el valor de la variable
; yaresultado en uno.
(assert (parametro modi))
(retract ?F)
)
```

```
(defrule rechaza
(declare (salience 9))
(tiene_todo)
(tiene numdef)
?F<-(tomo ?re1)
(tiene_re ?renor)
(test (>= ?re1 ?renor))
=>
(bind ?*solucion3* 1) ;se modifica el valor de la variable solucion3 con el carácter 1.
(bind ?*yaresultado* (+ ?*yaresultado* 1))
(assert (parametro modi))
(retract ?F)
)
```

```
(defrule cambia_rigu ;se define una regla y se le da un nombre.
(declare (salience 8))
?z<-(cambia_rigurosa)
=>
```

```
(bind ?*solucion3* CAMBIA A LA INSPECCIÓN RIGUROSA)
(retract ?z)
)
```

```
(defrule cambia_redu1
(declare (saliencia 8))
?q<-(diezsi_nummen)
;(elnumero_correcto)
=>
(bind ?*pregunta* INDIQUE SI LA PRODUCCIÓN ES ESTABLE:[SI/NO] )
(retract ?q)
)
```

```
(defrule cambia_redu2
(declare (saliencia 8))
?q<-(produccion_estable)
;(elnumero_correcto)
=>
(bind ?*pregunta* INDIQUE SI SE AUTORIZA CAMBIAR A REDUCIDA:[SI/NO])
(retract ?q)
)
```

```
(defrule cambia_redu3
(declare (saliencia 8)) ;se declara la prioridad de una regla.
?p<-(cambia_reducida)
;(elnumero_correcto)
=>
(bind ?*solucion3* CAMBIA A LA INSPECCIÓN REDUCIDA)
(retract ?p)
)
```

```
(defrule nocambia_redu
(declare (saliencia 7))
?d<-(nocambia_reducida)
=>
(retract ?d) ;se borra un hecho.
(assert (parametro modi)) ;se introduce un hecho
)
```

```
(defrule nocambia_redu1
(declare (saliencia 6)) ;antecedente de una regla.
?d<-(nocambia_reducida1)
=>
(retract ?d) ;consecuente de una regla.
(assert (parametro modi)))
```

```
(defrule yatieneredu
```

```

(declare (salience 15))
(cambia_reducida1)
?c1<-(tiene_aql0 ?aql11)
?c2<-(tiene_mmo ?mm11)
?c3<-(tiene_aco ?ac11)
(tiene_reo ?re11)
=>
(bind ?*solucion* Tiene una muestra aleatoria de ?mm11 piezas a inspeccionar)
(bind ?*solucion1* "El riesgo del productor es" ?aql11)
(bind ?*solucion2* Se aceptará el lote con ?ac11 se rechazará el lote con ?re11)
(assert (tiene_todore))
(assert (parametro modi1))
(retract ?c1)
(retract ?c2)
;(retract ?c3)
)
(defrule todore1
(declare (salience 14))
(cambia_reducida1)
(tiene_todore)
?E<-(parametro modi1)
=>
(bind ?*pregunta1* INDIQUE EL NÚMERO DE DEFECTUOSOS O DEFECTOS EN ESA
MUESTRA)
(assert (tiene numdef1))
(retract ?E)
)

(defrule acepre
(declare (salience 9))
(cambia_reducida1)
(tiene_todore)
?l<-(tiene numdef1)
?F<-(tomo1 ?re1)
(tiene_aco ?ac)
(tiene_reo ?re)
(test (< ?re1 ?re))
=>
(bind ?*solucion4* ?ac)
(bind ?*solucion5* ?re)
(bind ?*solucion3* 0)
(assert (pro_irregular))
(retract ?F)
(retract ?l)
)

(defrule rechazare

```



```
(declare (salience 9))
(cambia_reducida1)
(tiene_todore)
?l<-(tiene numdef1)
?F<-(tomo1 ?re1)
(tiene_aco ?ac)
(tiene_reo ?re)
(test (>= ?re1 ?re))
=>
(bind ?*solucion4* ?ac)
(bind ?*solucion5* ?re)
(bind ?*solucion3* 1)
(assert (parametro modi1))
(retract ?F)
(retract ?l)
)
```

```
(defrule produccion
(declare (salience 8))
(cambia_reducida1)
?k<-(pro_irregular)
=>
(bind ?*pregunta* SI LA PRODUCCIÓN ES IRREGULAR:[SI/NO])
(assert (otras_cond1))
(retract ?k)
)
```

```
(defrule condiciones
(declare (salience 7))
(cambia_reducida1)
?k1<-(otras_cond1)
=>
(bind ?*pregunta* ACONSEJAN OTRAS CONDICIONES CAMBIAR A NORMAL:[SI/NO])
(assert (parametro modi1))
(retract ?k1)
)
```

```
(defrule renormal
(declare (salience 15))
?w<-(cambia_reducida1)
?n<-(cambia_normal)
=>
(bind ?*solucion3* CAMBIA A LA INSPECCIÓN NORMAL)
(retract ?w)
(retract ?n)
(assert (tiene_todo))
(assert (parametro modi))
)
```

```
(assert (solo_p))
(bind ?*yaresultado* 0)
)
```

```
(defrule yatienerigu
(declare (saliencia 15))
(cambia_rigurosal)
?b1<-(tiene_aqloo ?aq11)
?b2<-(tiene_mmoo ?mm1)
?b3<-(tiene_acoo ?ac1)
(tiene_reoo ?ree)
=>
(bind ?*solucion* Tiene una muestra aleatoria de ?mm1 piezas a inspeccionar)
(bind ?*solucion1* "El riesgo del productor es" ?aq11)
(bind ?*solucion2* Se aceptará el lote con ?ac1 se rechazará el lote con ?ree)
(assert (tiene_todori))
(assert (parametro modi11))
(retract ?b1)
(retract ?b2)
;(retract ?b3)
)
```

```
(defrule todori1
(declare (saliencia 14))
(cambia_rigurosal)
(tiene_todori)
?E<-(parametro modi11)
=>
(bind ?*pregunta1* INDIQUE EL NÚMERO DE DEFECTUOSOS O DEFECTOS EN ESA
MUESTRA)
(assert (tiene numdef11))
(retract ?E))
```

```
(defrule acepri
(declare (saliencia 9))
(cambia_rigurosal)
(tiene_todori)
?l<-(tiene numdef11)
?F3<-(tomoloo_ya)
?F1<-(tomolo ?re11)
(tiene_acoo ?acc)
(tiene_reoo ?ree)
(test (< ?re11 ?ree))
=>
(bind ?*solucion4* ?acc)
(bind ?*solucion5* ?ree)
(bind ?*solucion3* 0)
```

```

(assert (parametro modil1))
(retract ?F1)
(retract ?F3)
(retract ?l)
)

(defrule rechazarii
(declare (saliencia 9))
(cambia_rigurosa1)
(tiene_todori)
?l<-(tiene numdefl1)
?F3<-(tomoloo_ya)
?F1<-(tomolo ?re11)
(tiene_acoo ?acc)
(tiene_reoo ?ree)
(test (>= ?re11 ?ree))
=>
(bind ?*solucion4* ?acc)
(bind ?*solucion5* ?ree)
(bind ?*solucion3* 1)
(assert (parametro modil1))
(retract ?F1)
(retract ?F3)
(retract ?l)
)
(defrule rinormal
(declare (saliencia 15))
?w<-(cambia_rigurosa1)
?n<-(cambia_normal)
=>
(bind ?*solucion3* CAMBIA A LA INSPECCIÓN NORMAL)
(retract ?w)
(retract ?n)
(assert (tiene_todo))
(assert (parametro modi))
(assert (solo_p))
(bind ?*yare resultado* 0)
)

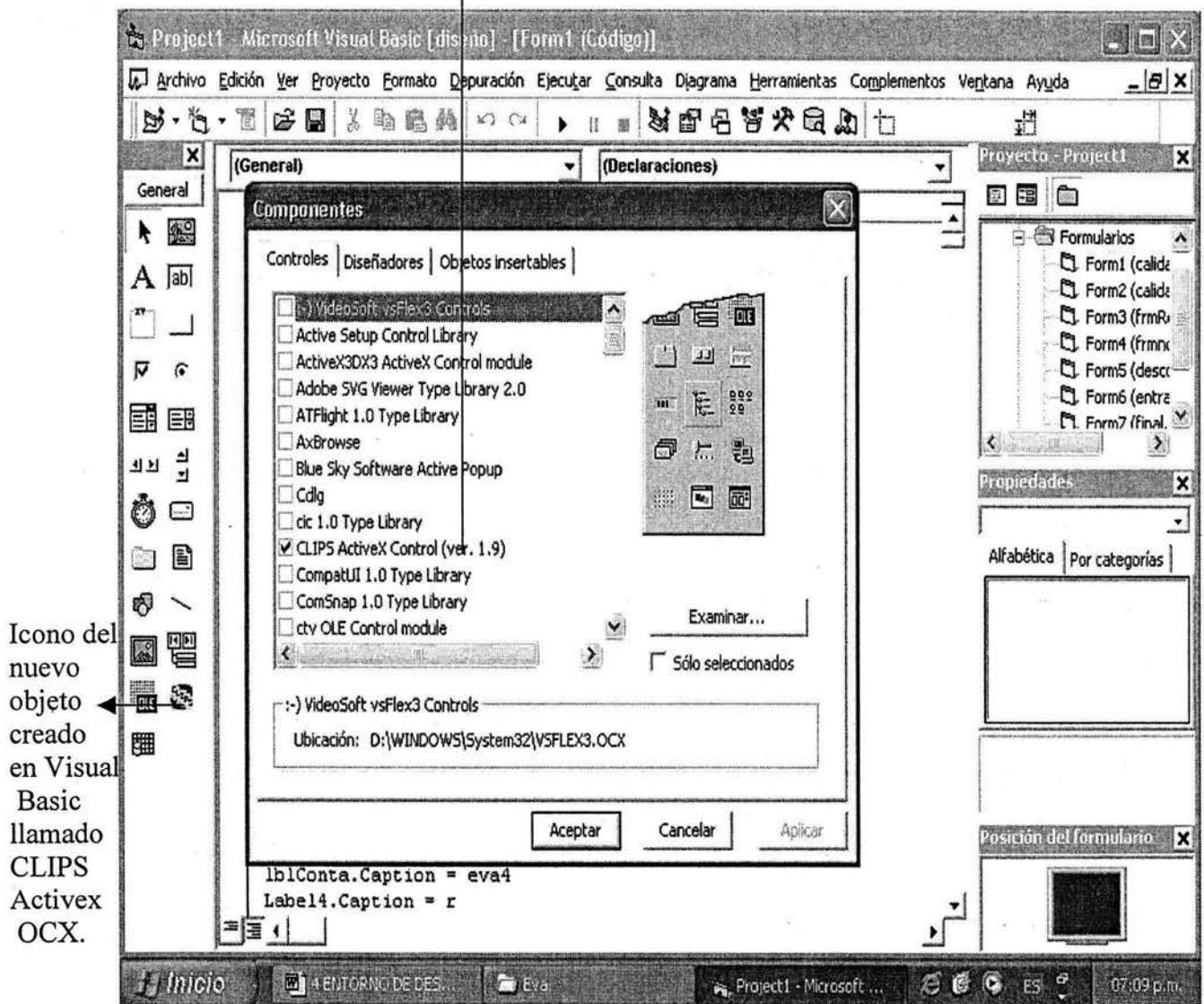
```

4.4 RELACIÓN ENTRE CLIPS, ACTIVEX OCX Y VISUAL BASIC.

NOTA: cuando se habla de Clips (en minúsculas) se hace referencia al shell o sistema experto llamado Clips y su entorno de trabajo, y cuando se habla de CLIPS (en mayúsculas) se hace referencia al nombre del objeto creado en Visual Basic anteriormente llamado CLIPSActiveXControl1 y renombrado como CLIPS.

Al instalar ActiveX OCX se crea en los componentes de Visual Basic un nuevo elemento llamado CLIPS ActiveX Control (ver.1.9) a su vez al seleccionar la casilla de verificación de este componente en Visual Basic se genera un nuevo objeto cuyo icono de identificación se puede observar directamente en la barra de herramientas como se ilustra en la figura 4.4.1 siguiente:

Nuevo componente
seleccionado llamado CLIPS
ActiveX control.



Para poder acceder a este objeto y a todos sus métodos y propiedades se coloca en una forma propia de Visual Basic un ejemplar u objeto de la clase CLIPS ActiveX OCX con nombre CLIPSActiveXControl1, propiedad que posteriormente se modificó en tiempo de diseño para renombrar el objeto CLIPSActiveXControl1 con un nombre más corto declarado como CLIPS.

Este objeto "CLIPS" hace llamadas a sus métodos y modifica sus propiedades en tiempo de ejecución, tales métodos o funciones son los comandos y sentencias de Clips vistos anteriormente en este capítulo.

Lo anterior se ilustra en un pequeño extracto del código de programación en Visual Basic en la figura 4.4.2.

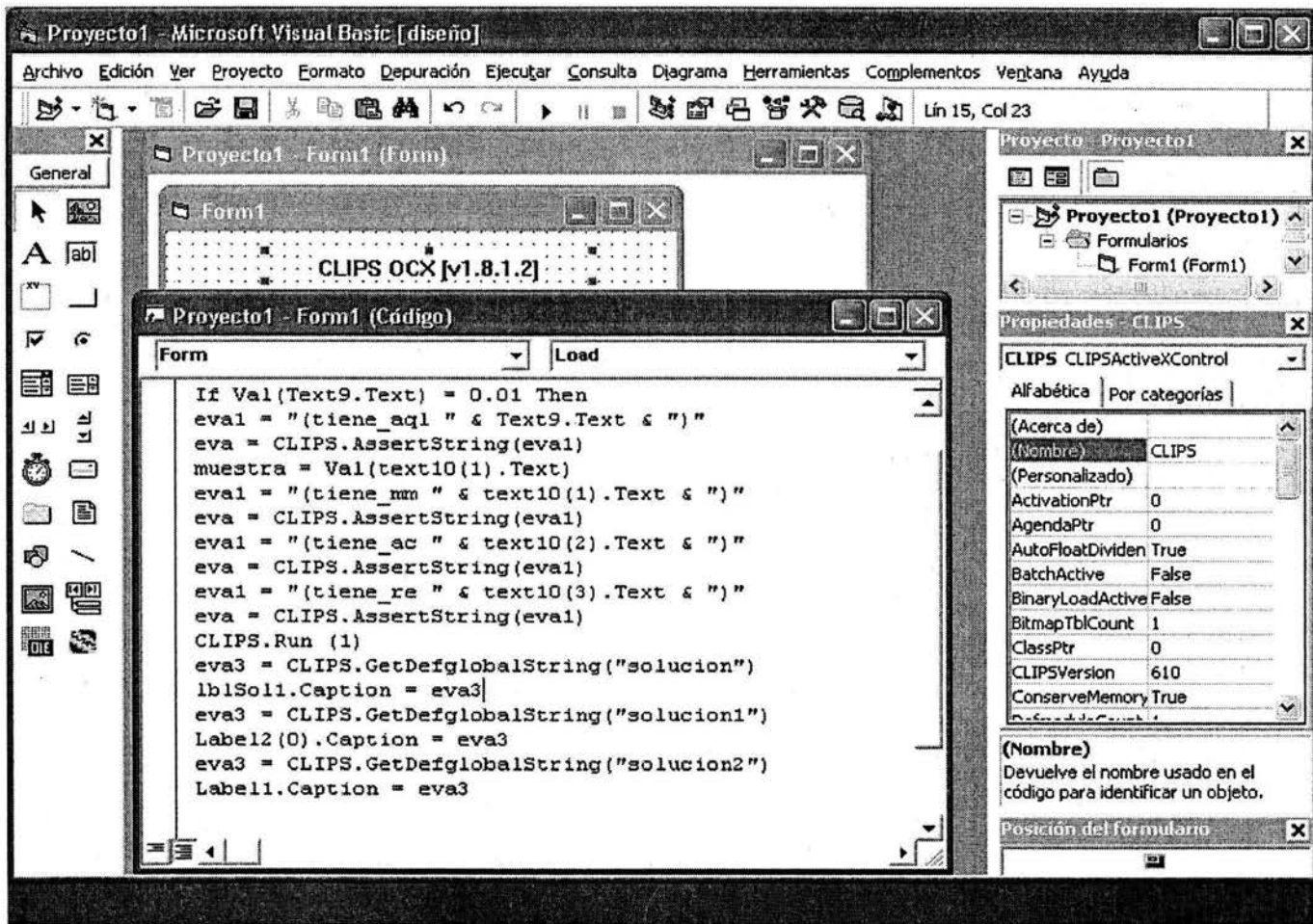


Figura 4.4.2

En la línea eval = "(tiene_aql " & Text9.Text & ")" de Visual Basic se observa que la cadena tiene_aql se concatena con el valor dado en el objeto de Visual Basic text9, tomado en su propiedad Text y asignando esta cadena de caracteres a una variable eval. En la línea eva = CLIPS.AssertString(eval) el objeto CLIPS de Visual Basic llama a su método AssertString que mete a la base de hechos del sistema experto el hecho referido en la variable eval la cual contiene una cadena de caracteres.

4.5 COMUNICACIÓN EXTERNA DEL SHELL CLIPS.

Para obtener o establecer una comunicación externa en Clips se hace referencia a la sentencia defglobal que declara variables globales, las cuales es necesario inicializarlas como Strings o cadenas de caracteres en blanco para después modificar su valor por medio de la sentencia bind y amarrarles una cadena de caracteres alfanuméricos que estructuran la pregunta o mensaje por medio del cual se puede obtener una comunicación visible en Visual Basic.

Abajo se muestra un extracto de la base de conocimiento del sistema experto para muestreo de aceptación.

```
(defglobal
 ?*pregunta* = "")
(defglobal
 ?*pregunta1* = "")
(defglobal
 ?*solucion* = "")

(defrule pedir
=>
(bind ?*pregunta* "PROPORCIONE EL TAMAÑO DEL LOTE" )
)

(defrule pediraql
(letra ?tama)
=>
(bind ?*solucion* la letra codificada es ?tama)
(bind ?*pregunta* "PROPORCIONE EL RIESGO DEL PRODUCTOR" )
(assert (solo_p)))
```

Posteriormente el objeto CLIPS a través de su método o función GetDefglobalString que recibe como parámetro una cadena, que en este caso es la variable global llamada ("solucion") accede a los mensajes de la base de conocimiento del sistema experto en Clips, y los asigna a una variable llamada eva3.

El contenido de eva3 será mostrado después en otro objeto de Visual Basic el lblSol1 modificando su propiedad Caption en tiempo de ejecución, de esta forma los mensajes o cadenas de caracteres “pregunta y soluciones” de la base de conocimiento del sistema experto pueden ser visibles en Visual Basic.

Lo anterior se ejemplifica en los párrafos siguientes que son líneas de programación en Visual Basic.

```
CLIPS.Run (1)
eva3 = CLIPS.GetDefglobalString("solucion")
lblSol1.Caption = eva3
```

```
eva3 = CLIPS.GetDefglobalString("solucion1")  
Label2(0).Caption = eva3  
eva3 = CLIPS.GetDefglobalString("solucion2")  
Label1.Caption = eva3  
CLIPS.Run (1)  
eva3 = CLIPS.GetDefglobalString("pregunta")  
lblPreg.Caption = eva3  
Text9.Text = ""
```

4.6 ANÁLISIS DEL SISTEMA.

El término análisis aplicado a sistemas, significa descomponerlos en sus componentes, para estudiar cada uno de ellos, tanto como un entidad aislada, como en interacción con el resto. Para ser útil, al análisis le debe seguir la síntesis, que consiste en unir los componentes del sistema para ver como funcionan en conjunto.

El análisis del sistema es una actividad que consiste en transformar la especificación de los requerimientos en modelos que representan datos y las transformaciones que realiza el sistema sobre estos.

Los DFD (Diagramas De Flujo De Datos) es una técnica que se utiliza principalmente para la modelización de sistemas informáticos. Representa el flujo de la información, las transformaciones que se aplican y los datos al moverse desde la entrada a la salida en un sistema.

Los DFD representan las transformaciones que sufren los datos, los datos que se transfieren entre los procesos del sistema, la información que se almacena y los productores o receptores de información.

Las representaciones gráficas de la funcionalidad del sistema mediante los DFD deben completarse utilizando documentos descriptivos desarrollados adecuadamente que proporcionen información sobre los datos, los almacenes y las especificaciones de los procesos.

Los DFD nos permiten visualizar al sistema experto para muestreo de aceptación como una red de procesos funcionales que describen las transformaciones que sufren los datos que se le proporcionan.

4.6.1 DFD DEL SISTEMA DE NIVEL 0.

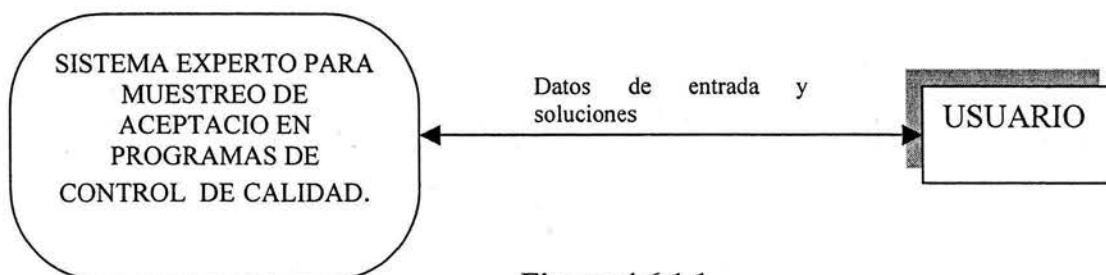


Figura 4.6.1.1

En la figura 4.6.1.1 se muestra el diagrama de flujo de datos para el sistema en general, en el cual el sistema se nutre de los datos que le pide a un ente externo que es el usuario y estos datos son procesados y transformados para que el sistema le de al usuario la solución respecto el problema específico para el cual fue diseñado éste sistema experto.

4.6.2 DFD DEL SISTEMA DE NIVEL 1.

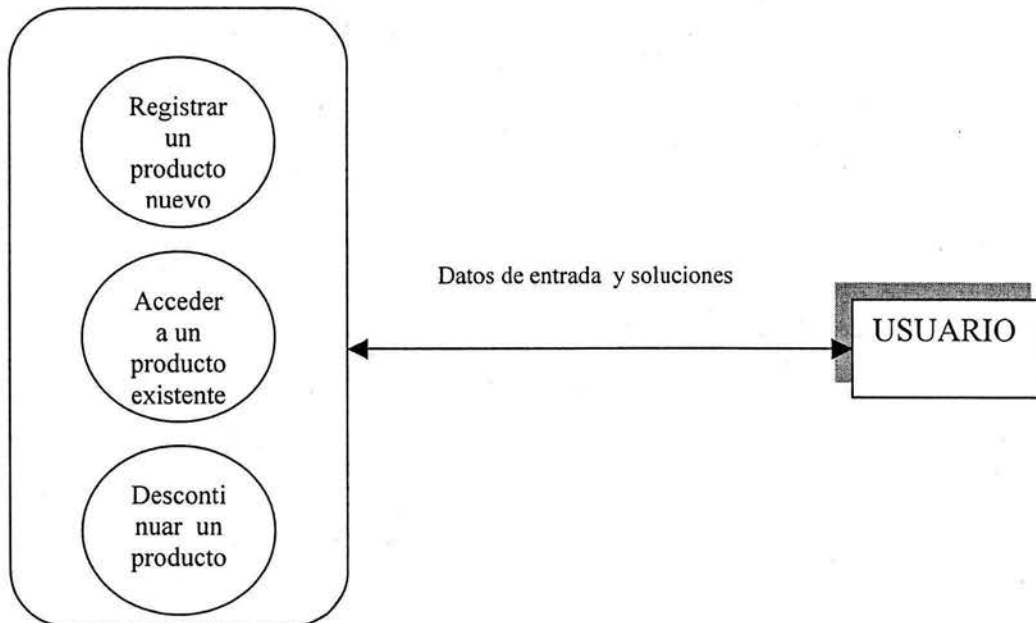


Figura 4.6.2.1

En el DFD del sistema del nivel 1 que se muestra en la figura 4.6.2.1 se identifican tres principales procesos el primero: “registrar un producto nuevo”, es el primer actor debido a que este proceso permite dar de alta un producto nuevo dentro del sistema en un espacio libre en su base de datos, es decir actualiza los datos de la base de datos con valores apropiados diferentes de nulo y cero, el proceso toma estos datos de la base de datos y los ve o transforma como una información global es decir como un producto ya capturado o registrado y no como datos independientes aislados.

El siguiente proceso: “acceder a un producto existente” permite al usuario acceder a un producto que tiene datos aceptados por el sistema en su base de datos, es decir el producto ya existe o esta capturado en el sistema con valores de datos apropiados. Este proceso actualiza en la base de datos el número de entidades defectuosas de ese producto y modifica la base de datos con la última información proporcionada al sistema para transformar estos datos en una solución de cambio de inspección como normal, reducida, rigurosa o inspección descontinuada.

Se debe recordar que este sistema puede únicamente trabajar con tres productos, dicho de otra forma su base de datos consta únicamente de tres registros o renglones correspondientes al producto uno, producto dos y producto tres. En esta tesis se adoptó esta limitación para poder verificar el funcionamiento de las estructuras necesarias, pero en la realidad esta limitación se puede eliminar.

El último proceso: “Descontinuar un producto” se utiliza para mantenimiento del sistema ya que da de baja del sistema un producto existente. Este proceso borra todos los datos anteriormente dados al sistema y pone valores de nulo y cero en los registros pertenecientes a ese producto en la base de datos liberando del sistema ese espacio ocupado antes por un producto y dejándolo para poder ser utilizado por otro producto que se quiera capturar o registrar después.

4.6.3 DFD DEL SISTEMA DE NIVEL 2 PARA EL PROCESO “REGISTRAR UN PRODUCTO NUEVO”.

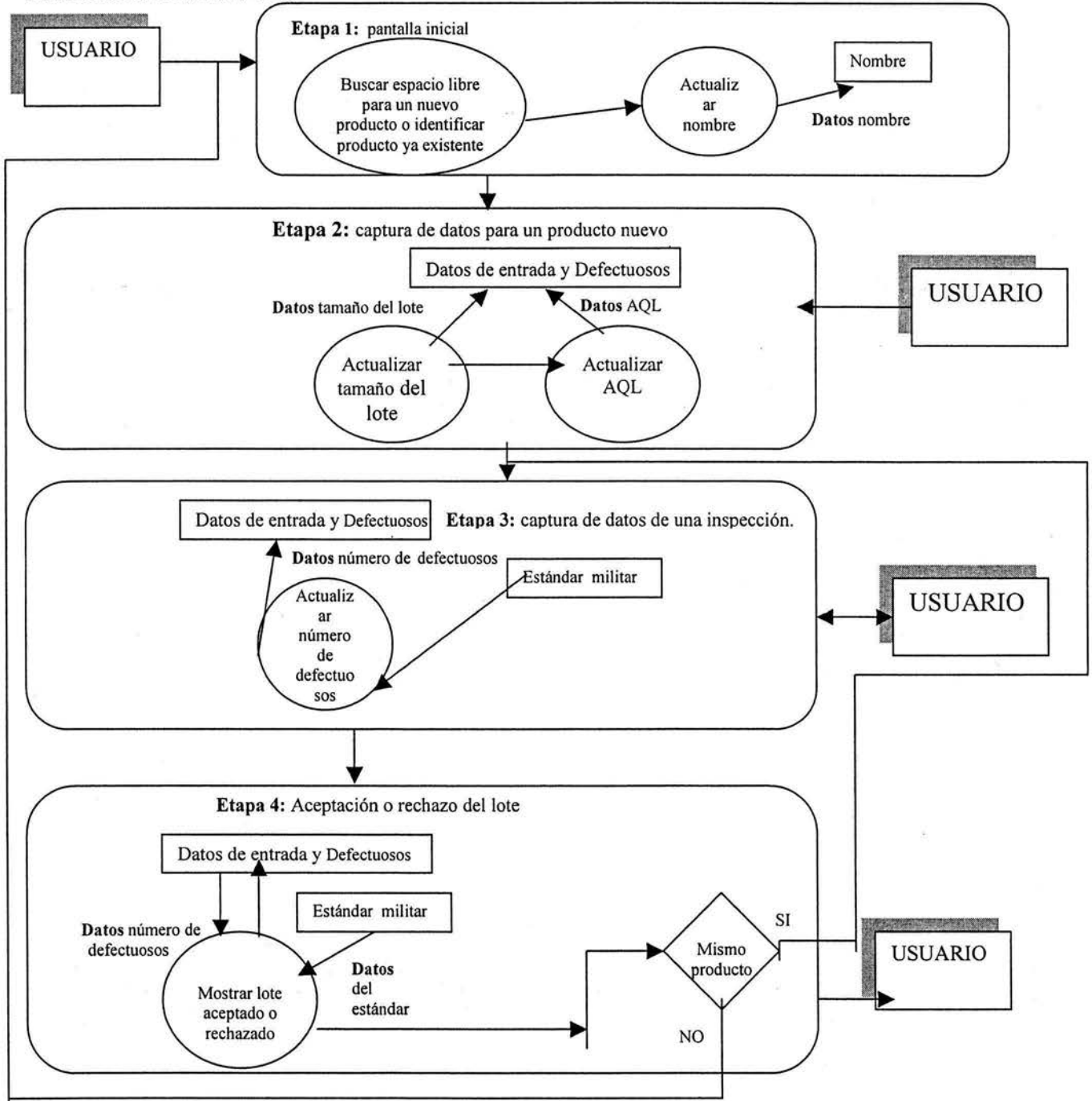


Figura 4.6.3.

En la figura 4.6.3.1 se muestra el DFD del sistema de nivel 2 para el proceso “registrar un producto nuevo”, el cual comienza por un proceso inicial “Buscar espacio libre para un nuevo producto o identificar producto ya existente” que recibe eventos de parte del usuario y los transforma en pantallas amigables o mejor dicho en una interfase grafica de usuario con la que interactúa éste ente externo.

Existen dos caminos en a seguir: el proceso buscar un espacio libre en el sistema para poder registrar un producto nuevo o el proceso seleccionar un producto existente registrado ya en el sistema.

Para el proceso “registrar un producto nuevo”, el usuario genera el evento para que el sistema busque un espacio libre y pueda dar de alta un nuevo producto, activando una caja de dialogo en el siguiente proceso llamado “actualizar nombre”, este proceso interactúa con el usuario quien proporciona un nombre para dicho producto nuevo. Estos datos al salir de éste proceso son transformados en una etiqueta de identificación para ese producto en todas las etapas del sistema y son guardados en un archivo o base de datos.

Esta información entra en la siguiente etapa (etapa 2) del sistema que contiene dos procesos los cuales se encargan de actualizar la información principal para que el sistema pueda trabajar, el primer proceso “actualizar tamaño del lote” actualiza el tamaño del lote y el segundo proceso “actualizar AQL” actualiza el riesgo del productor o AQL. Estos datos son proporcionados por el usuario y son actualizados en los archivos del sistema.

En la siguiente etapa (etapa 3) del sistema el proceso trabaja con la información anterior y está listo para recibir el número de unidades defectuosas de los diferentes lotes de este producto que el usuario le proporcionará, datos que el proceso “actualizar número de defectuosos” se encarga de actualizar en la base de datos del sistema además de mostrar al usuario información necesaria para esta inspección, obtenida de otro archivo o base de datos también del sistema, que contiene las especificaciones del estándar militar que se utilizó como línea de desarrollo.

En la siguiente etapa (etapa 4) el proceso “mostrar lote aceptado o rechazado” obtiene información de las dos bases de datos antes referidas para dar un parámetro de comparación entre el número limite de entidades defectuosas para poder extraer una conclusión respecto a la aceptación o rechazo del lote, conclusión que el sistema experto muestra al usuario cada vez que se pone en marcha este proceso.

El usuario obtiene esta información llamada solución de segundo tipo y al final de cada solución o conclusión dada, se visualiza un elemento de decisión que se refiere a si el usuario quiere seguir con la inspección de ese mismo producto o si el usuario prefiere regresar a la etapa inicial (etapa 1), donde ahí podrá seleccionar cualquiera de las dos alternativas antes mencionadas (buscar espacio libre para un nuevo producto o identificar producto ya existente).

La figura 4.6.4.1. muestra el DFD del sistema de nivel 2 para el proceso “acceder a un producto existente”.

En la primera etapa (etapa 1) el usuario genera el evento para decidir entre tres posibles productos, producto uno, producto dos, o producto tres, etiquetados con un nombre característico que distingue a cada producto, esto si los tres espacios anteriores han sido dados de alta tiempo atrás y están habilitados, pero puede darse el caso de que existan solo dos, uno o que no exista ningún producto dado de alta o habilitado. Si no hay productos dados de alta se tendrá que iniciar el sistema en el proceso “registrar un producto nuevo”.

Si existen productos dados de alta y el usuario quiere acceder a uno de ellos, este interactúa con una interfase grafica en el proceso “Buscar espacio libre para un nuevo producto o identificar producto ya existente” para indicarle al sistema que lo que necesita es seleccionar un producto ya existente para seguir su inspección anteriormente detenida.

El siguiente proceso llamado “Identificación del producto existente” se encarga de seleccionar de la lista el producto electo por el usuario y pasar los datos de ese producto a la siguiente etapa (etapa 3) desapareciendo la etapa 2, la cual solo se activa para el proceso “registrar un producto nuevo”.

Este producto anteriormente pasó por ésta etapa (etapa2) y tomo los datos necesarios y adecuados para poder ser visto como un componente perteneciente al sistema, el cual fue resguardado en su base de datos.

En esta etapa (etapa 3) el proceso “actualizar número de defectuosos” trabaja con los datos de ese producto antes guardados en la base de datos y actualiza estos datos con los proporcionados por el usuario para pasarlos a la siguiente etapa (etapa 4) además de que muestra al usuario información necesaria para la inspección, obtenida de la base de datos donde se archivan las definiciones y los datos del estándar militar que fue objeto de estudio.

En la etapa (etapa 4) los datos son transferidos al siguiente proceso “mostrar cambio o discontinuó de inspección” ésta etapa (etapa 4) trabaja en forma similarmente antes descrita en el capítulo 4.6.3.

El proceso “mostrar lote aceptado o rechazado” obtiene información de las dos bases de datos del sistema para dar un parámetro de comparación entre el número límite de entidades defectuosas para poder extraer una conclusión respecto a la aceptación o rechazo del lote, conclusión que el sistema experto muestra al usuario cada vez que se pone en marcha este proceso.

Al final de esta conclusión sobre la aceptación o rechazo del lote se visualiza un elemento de decisión, el cual pregunta al usuario si este desea regresar a la etapa inicial (etapa 1) para buscar un espacio libre para un nuevo producto o identificar producto ya existente, o bien desea continuar con esta misma inspección.

Si se sigue en esta línea con la misma inspección los datos son pasados el proceso “mostrar cambio de régimen de inspección”.

Este proceso ordena los datos de la base de datos para ver si alguna regla de la base de conocimiento de Clips que actúa como un ente externo al sistema puede ser disparada y así poder dar un resultado al usuario en cuanto al cambio de régimen de inspección (solución de primer tipo), por ejemplo para el cambio de inspección de normal a reducida el sistema además preguntara al usuario si la producción es estable y si el funcionario autorizado acepta dicho cambio y se mostrara una confirmación a este cambio.

El sistema también podrá mostrar al usuario únicamente información respecto a sí el lote es aceptado o rechazado (solución de segundo tipo) sin obtener información en cuanto al cambio de régimen de inspección. Esto último en los casos en los que el estándar militar no contemple un cambio de régimen de inspección.

Si se obtienen soluciones de primero y segundo tipo, no aparecerá ese elemento de decisión y el sistema arrojará los dos tipos de soluciones y un elemento de confirmación al cambio de régimen de inspección una vez acreditado este cambio, aparecerá el elemento de decisión para seguir en el mismo sentido (a la etapa 3) con el mismo producto o regresar a la etapa inicial (etapa 1) para acceder a otro producto o dar de alta uno nuevo.

4.6.5 DFD DEL SISTEMA DE NIVEL 2 PARA EL PROCESO “DESCONTINUAR UN PRODUCTO”.

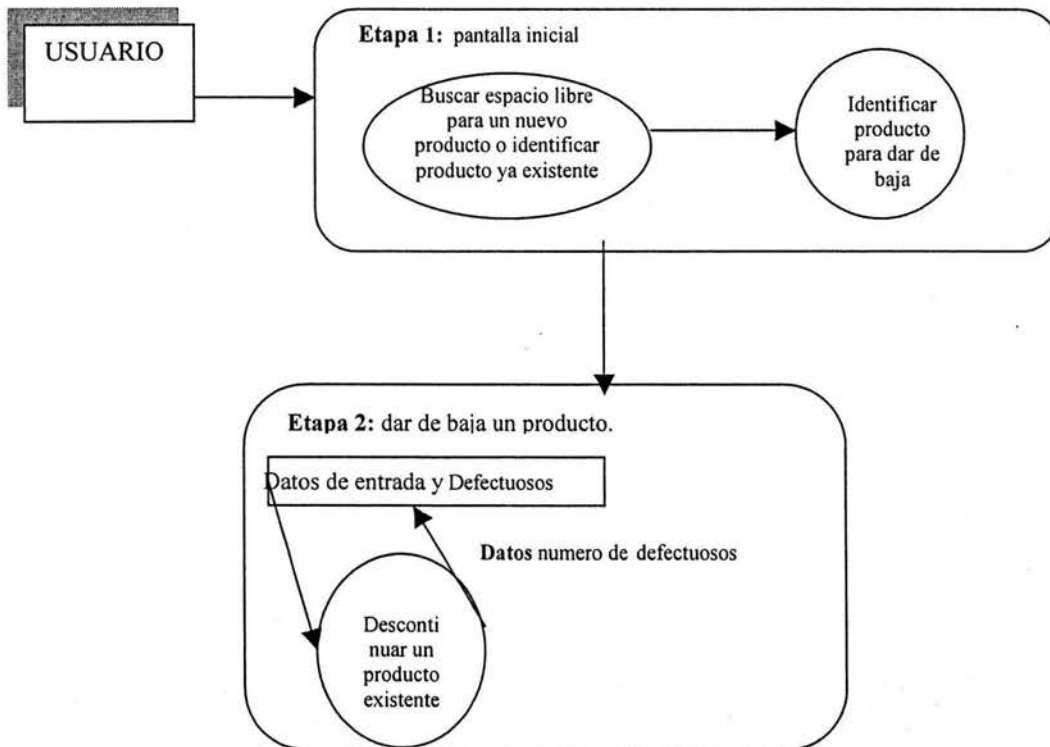


Figura 4.6.5.1

La figura 4.6.5.1 muestra el DFD del sistema de nivel 2 para el proceso “descontinuar un producto”. Este proceso se encarga de dar de baja del sistema un producto capturado anteriormente con datos apropiados para que el sistema pueda ver al producto como un ente global y trabajar con él como un producto etiquetado con un nombre que lo identifica y no solo contener o capturar datos y parámetros aislados independientes que el sistema no registrara y por lo tanto no podrá darle al usuario una solución al respecto.

En la primera etapa (etapa 1) el usuario genera los eventos necesarios en el proceso “Buscar espacio libre para un nuevo producto o identificar producto ya existente” para que el sistema le presente una interfase grafica al usuario contenida en el proceso “Identificar producto para dar de baja” que muestra al usuario tres productos diferentes que podrá dar de baja de dicho sistema, si fuera el caso de que los tres archivos estén ocupados por algún producto o ente global, pero puede haber tres, solo dos o únicamente un elemento registrado en el sistema, el cual o de entre los cuales se decidirá el producto candidato a darse de baja.

Una vez elegido dicho producto para darlo de baja los datos pasan a la etapa siguiente (etapa 2) que contiene el proceso “descontinuar un producto existente” en el cual el sistema obtiene de la base de datos la información acreditada por el sistema de ese producto y la actualiza por valores de nulo y cero dejando dentro del sistema un espacio libre en el cual se podrá ingresar un nuevo producto, debido a que el anterior producto que se dio de baja quedará completamente borrado del sistema.

CAPITULO 5

EJEMPLOS DE LA APLICACIÓN DEL SISTEMA EXPERTO

El objetivo de este capítulo es representar gráficamente una aplicación del sistema experto a un problema específico y mostrar las secuencia de pantallas que irán apareciendo en el camino para llegar a una solución, describiendo cada uno de los elementos que existan en esta interfase de comunicación.

5.1 PANTALLA PRINCIPAL.

La pantalla principal es el punto de entrada para el usuario y en ésta el usuario puede escoger entre empezar o seguir con una inspección, además puede seleccionar la historia de un producto existente o si prefiere, dar de baja un producto.

Esta pantalla muestra una ilustración de una distribución binomial que se emplea para resolver problemas estadísticos y en la parte inferior derecha se ven tres botones con los nombres “CONSULTA”, “MANTENIMIENTO” e “HISTORIA”. Esta pantalla se finaliza con un botón de SALIR que aparece mas abajo. La figura 5.1.1 ilustra la pantalla principal.

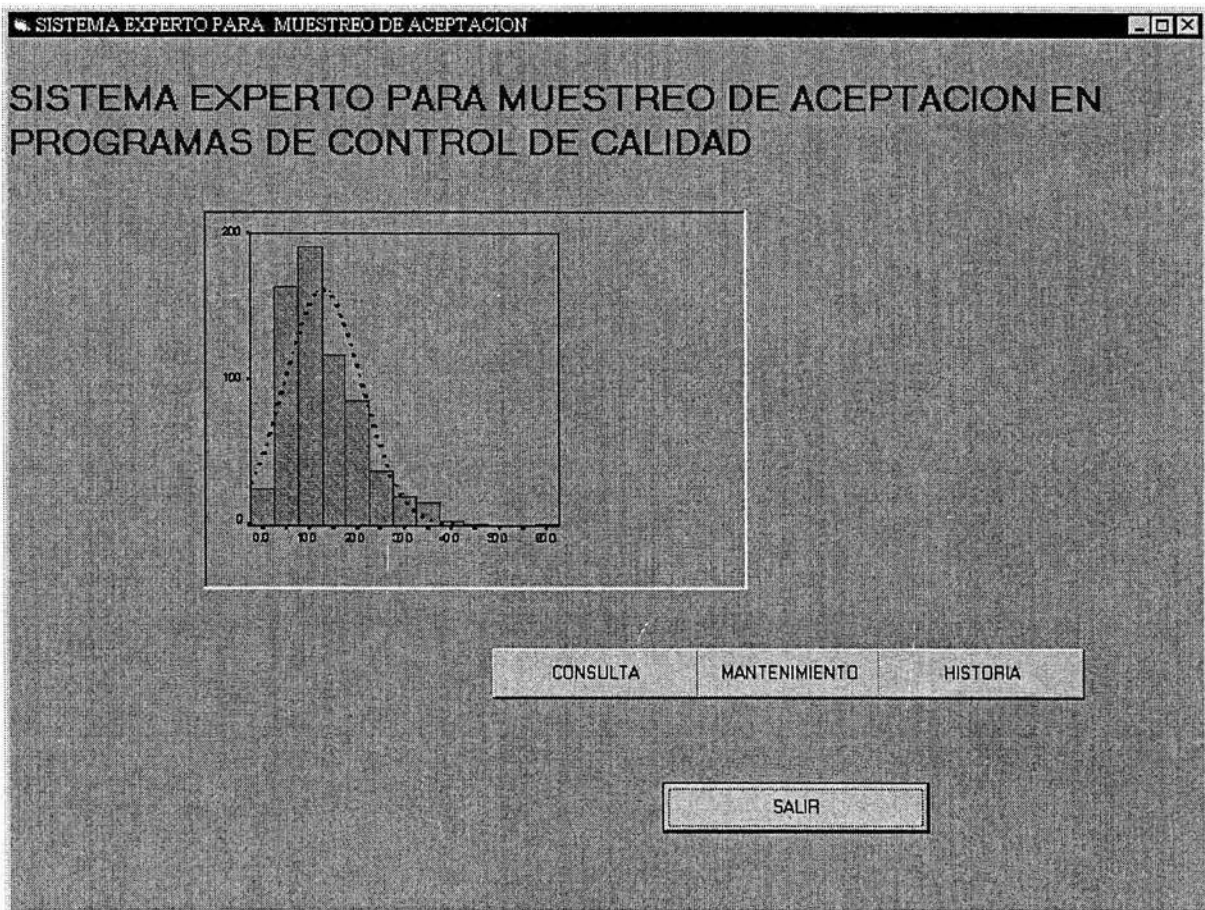


Figura 5.1.1

Al seleccionarse el botón MANTENIMIENTO se muestra otra pantalla en donde se indica que productos están capturados en el sistema, de los que se puede elegir cual de ellos dejará de estar bajo inspección para borrarlo completamente, dejando espacio para registrar un nuevo producto.

La figura 5.1.2, muestra únicamente dos botones habilitados para darse de baja, con los nombres “TUERCAS” y “RONDANAS”.

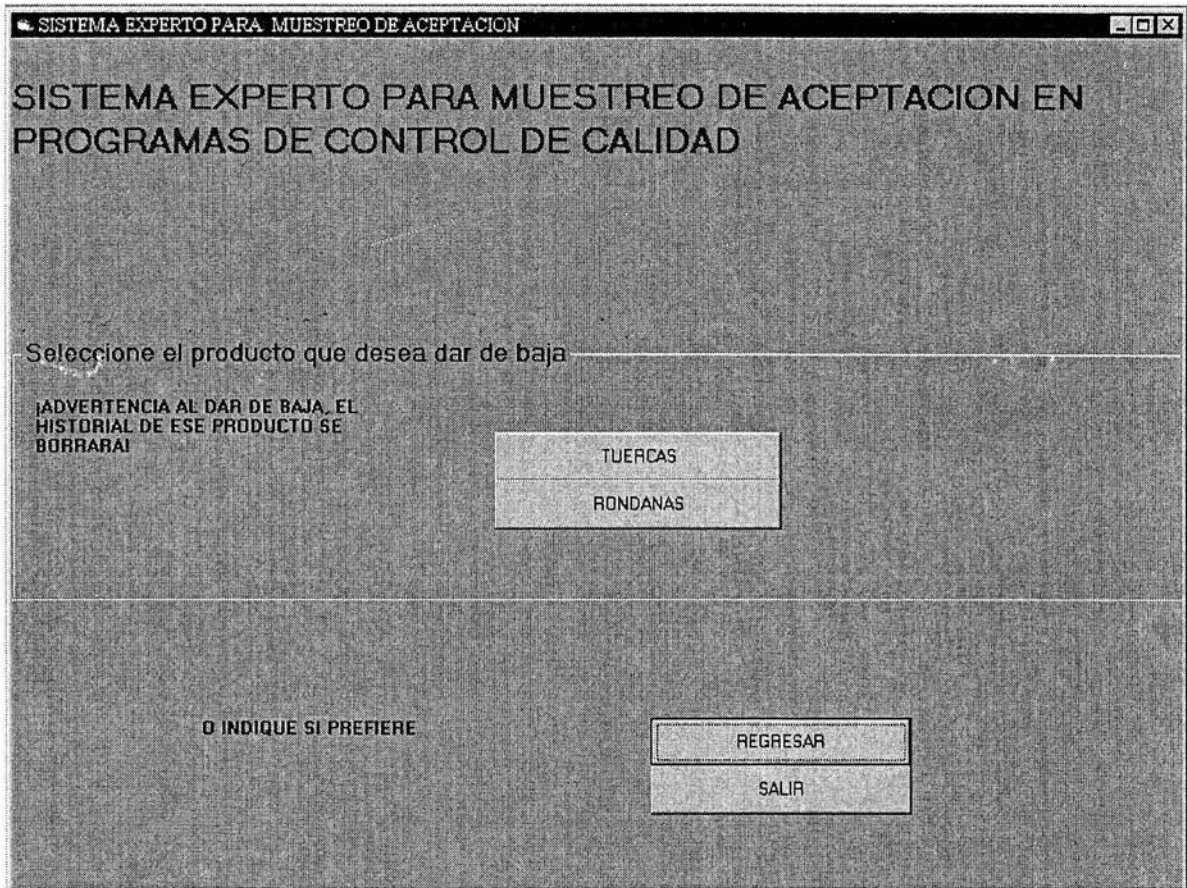


Figura 5.1.2

Dado que las “RONDANAS” no seguirá con su inspección, se darán de baja seleccionando el botón que tiene el nombre de éste producto con el ratón.

Después de ésta acción aparecerá en pantalla un cuadro de mensaje para confirmar si el producto seleccionado se dará de baja, como lo ilustra la figura 5.1.3. Si se le indica al sistema que si se quiere la baja, aparecerá otro cuadro de mensaje avisando que el producto seleccionado en este caso el producto “RONDANAS” han sido eliminado como se observa en la figura 5.1.4. y si se le indica al sistema que no se lleve a cabo la baja, el sistema volverá a la pantalla anterior (figura5.1.2).

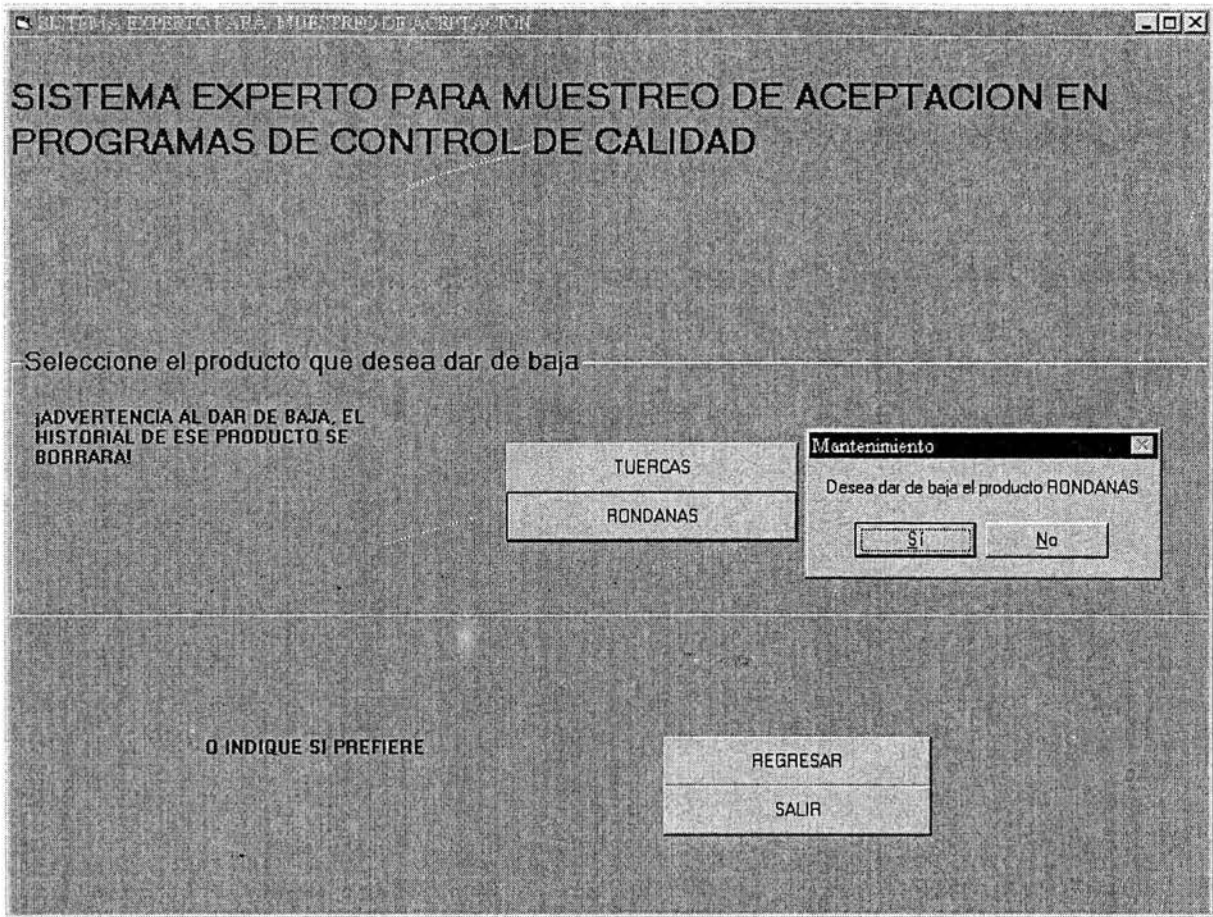


Figura 5.1.3

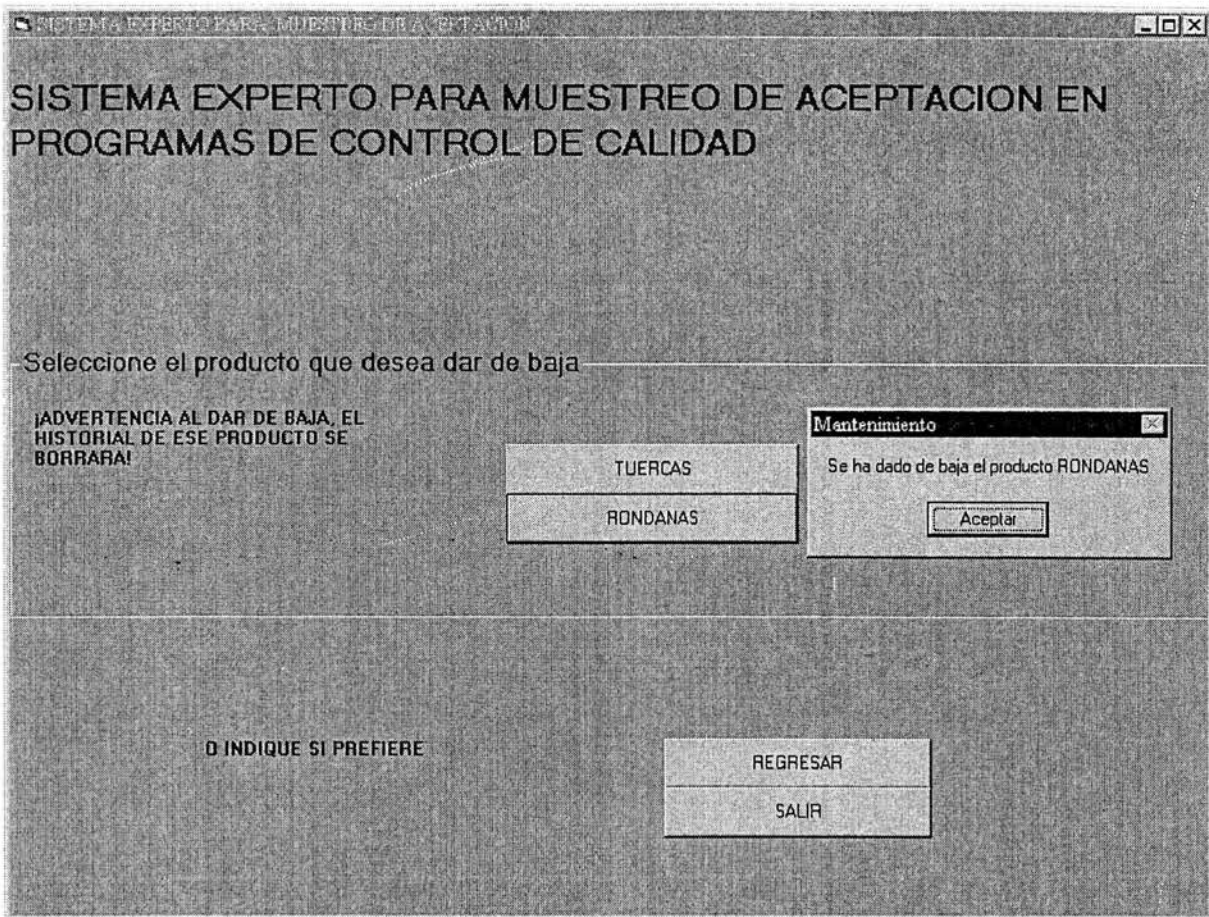


Figura 5.1.4

Al aceptar en este mensaje aparecerá la pantalla siguiente o figura 5.1.5 en donde se observa que han desaparecido las “RONDANAS”.

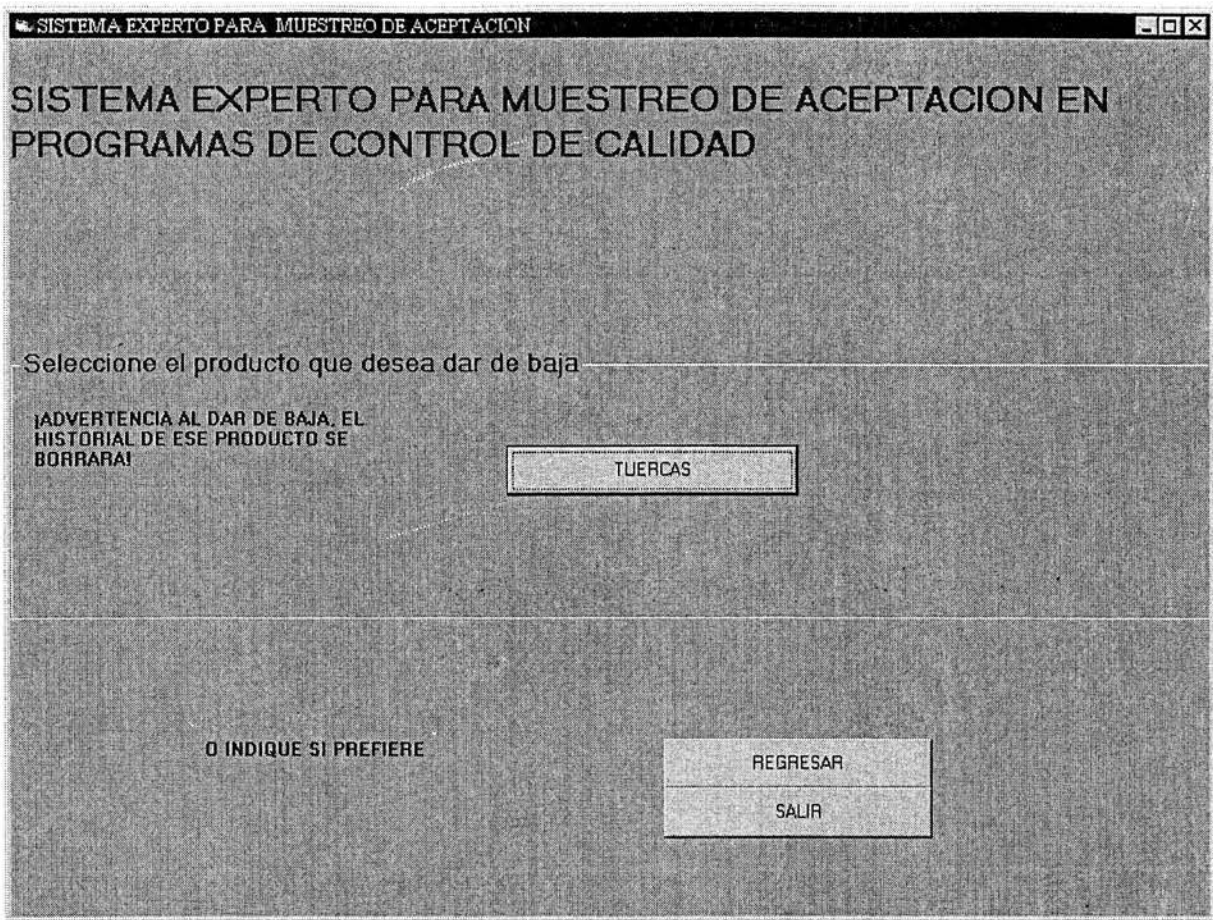


Figura 5.1.5

Los botones REGRESAR y SALIR aparecen en todas las pantallas antes mostradas, el botón REGRESAR, nos regresa a la pantalla inicial (figura5.1.1) y el botón SALIR termina la aplicación.

Debido a que el producto “TUERCAS” está aun capturado en el sistema tiene una historia que hace referencia a toda su inspección que lleva hasta el momento. Para consultar la historia de un producto capturado se selecciona el botón HISTORIA de la pantalla principal (figura5.1.1). Esto dará lugar a que aparezca otra pantalla mostrada en la figura 5.1.6 que señala los productos que tienen historia es decir los que están capturados, dado que los productos borrados han desaparecido y también su historia dejando lugar para capturar un nuevo producto.

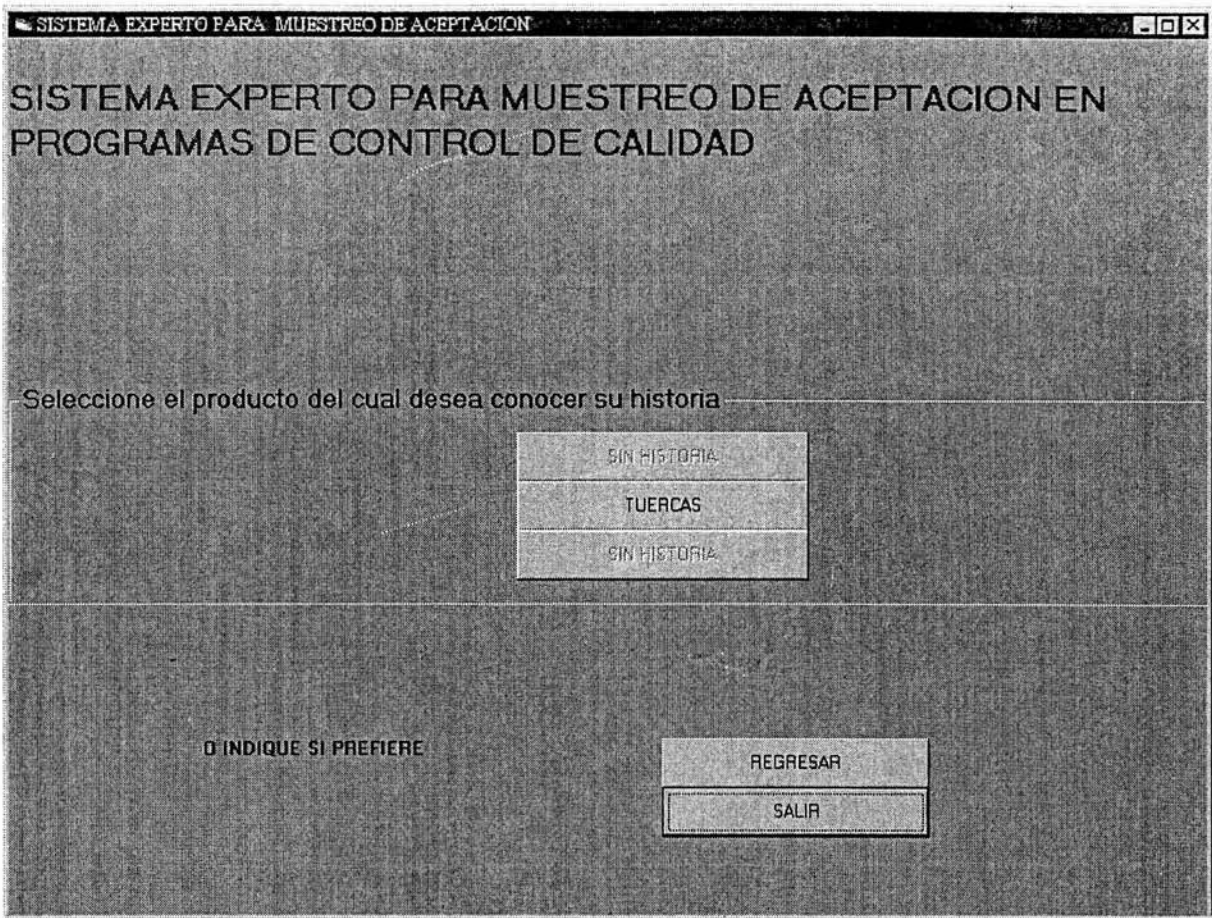


Figura 5.1.6

Como se observa en la figura 5.1.6 existen dos botones deshabilitados, es decir que hay dos lugares vacíos en el sistema en donde podremos ingresar dos productos nuevos, como se menciona en el capítulo anterior (capítulo 4) este sistema solo cuenta con tres lugares para registrar tres productos y en esta pantalla se puede ver que un lugar está ocupado por el producto "TUERCAS" y se hallan dos lugares libres.

Se consulta la historia del producto "TUERCAS" seleccionando su botón que está habilitado para conocer su historia, al seleccionar este botón aparecerá la historia del producto "TUERCAS" tal como lo muestra la figura 5.1.7.

Además en la figura 5.1.7 se presenta el botón REGRESAR A LA PANTALLA INICIAL y el botón SALIR DEL SISTEMA. Con el primero regresamos a la pantalla inicial o figura 5.1.1 y con el segundo salimos del sistema.

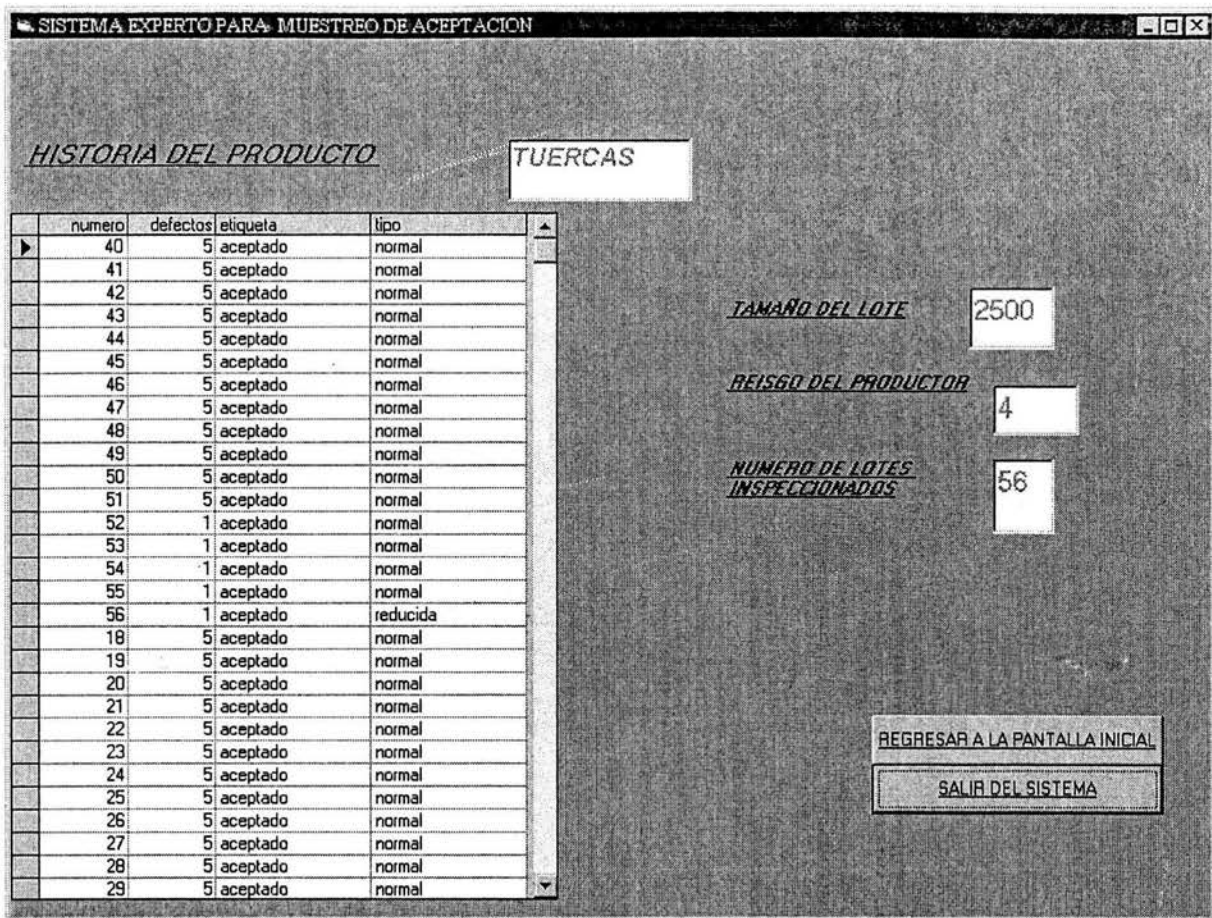


Figura 5.1.7

Para continuar o empezar con la inspección de un producto se selecciona el botón llamado CONSULTA que aparece en la pantalla principal (figura 5.1.1). Este botón mostrará otra pantalla que contiene dos botones más como lo ilustra la figura 5.1.8.

Con el botón “con historial existente” se indica al sistema que prosiga con la inspección de un producto anteriormente dado de alta. La figura 5.1.9 aparece después de seleccionar el botón “con historial existente” y contiene la lista de aquellos productos que tienen un historial.

ESTA TESIS NO SALE
 DE LA BIBLIOTECA

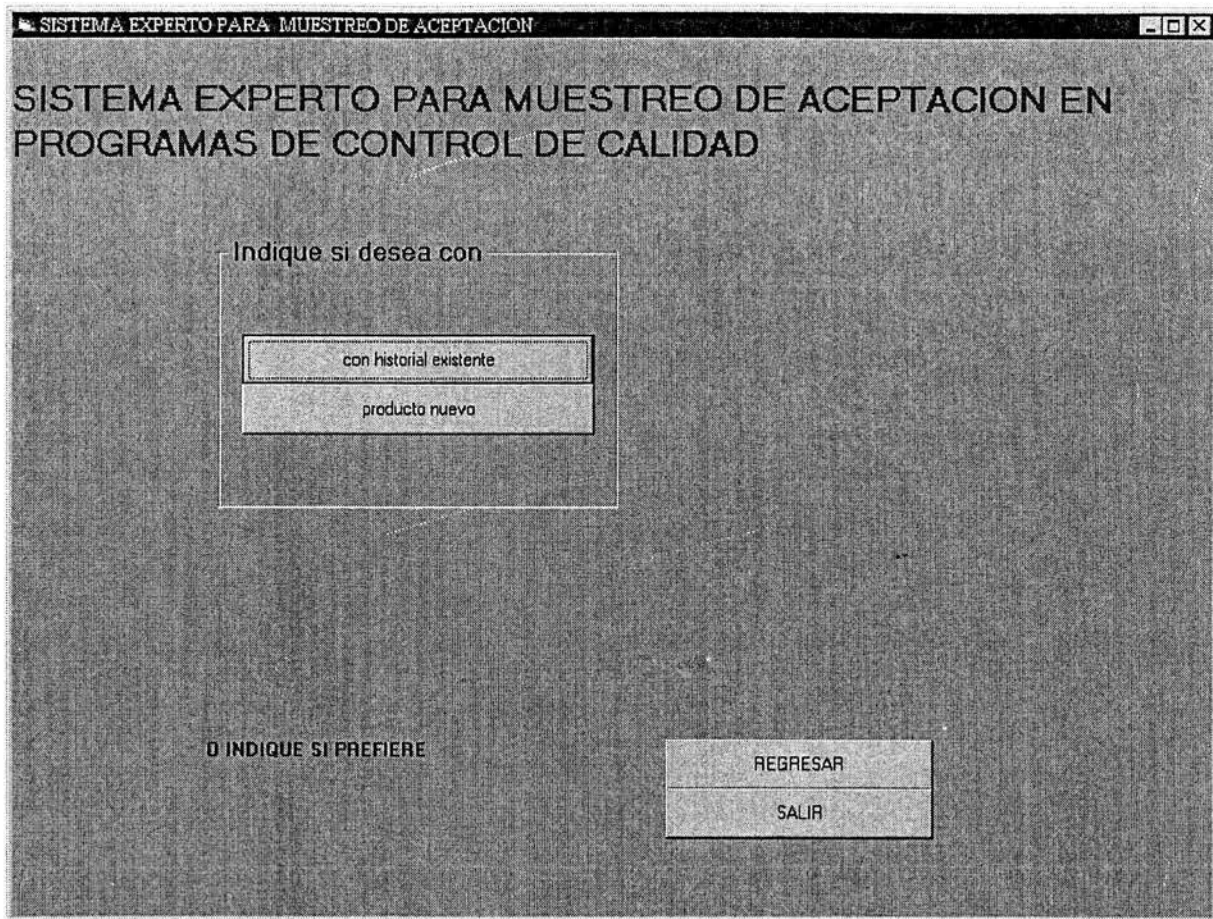


Figura 5.1.8

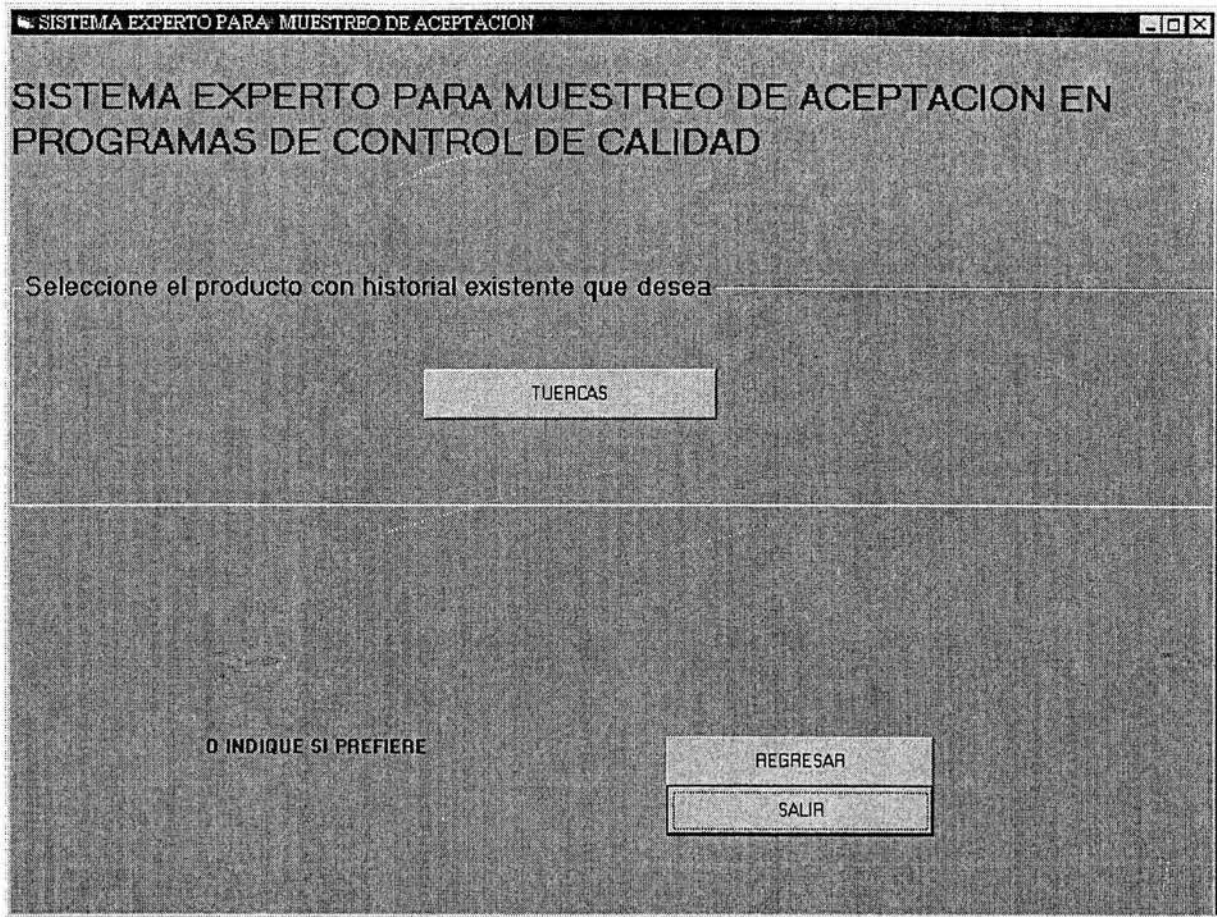


Figura 5.1.9

Al seleccionar el producto deseado con historial existente se continúa con su inspección debido a que el sistema experto nos conduce a la pantalla en la cual anteriormente el producto se había finalizado, es decir regresamos a la inspección en la cual se había quedado ese producto en él último estudio del sistema y que puede ser normal, reducida o rigurosa.

En este caso se prosigue con la inspección del producto "TUERCAS" como lo muestra la figura 5.1.10 este producto continuará en una inspección normal.

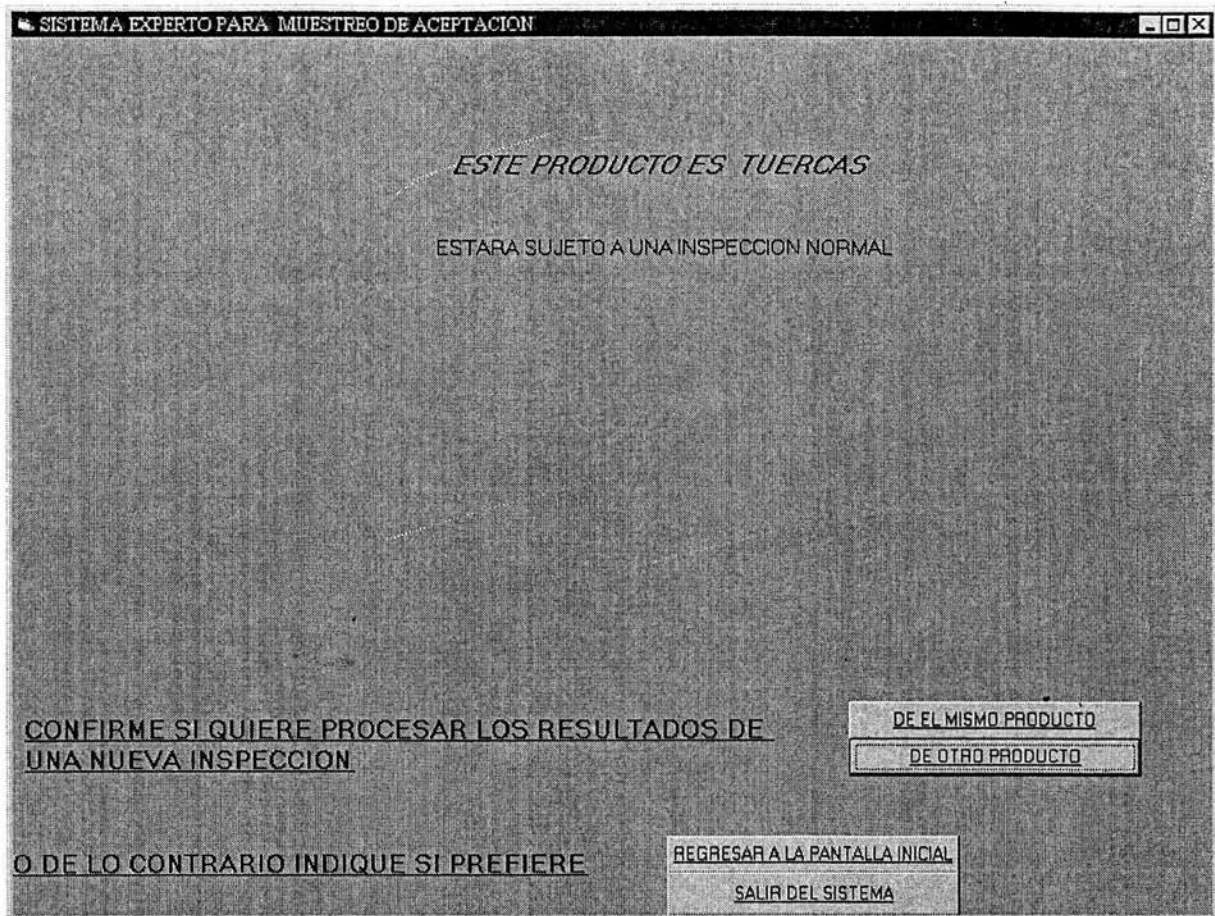


Figura 5.1.10

El botón DEL MISMO PRODUCTO conduce a la captura de datos para el producto seleccionado cuyo nombre se puede ver en la leyenda al principio de la pantalla que dice ESTE PRODUCTO ES “nombre del producto” en este caso son las “TUERCAS” entonces esta línea se puede leer como “ESTE PRODUCTO ES TUERCAS” y está ilustrada en la figura anterior (figura5.1.10).

Se selecciona el botón DE OTRO PRODUCTO para regresar a la pantalla mostrada en la figura 5.1.8, si por alguna razón de último momento se decide que no se quiere continuar con la inspección del producto seleccionado y se desea elegir otro producto para continuar su inspección, o capturar un nuevo producto.

Se selecciona el botón REGRESAR A LA PANTALLA INICIAL para que se muestre la pantalla inicial de la figura 5.1.1 en la cual se puede revisar la historia de algún producto o dar mantenimiento al sistema.

La figura 5.1.11 muestra la pantalla en donde se capturan los resultados de una inspección para proseguir con ésta después de seleccionado el botón DE EL MISMO PRODUCTO.

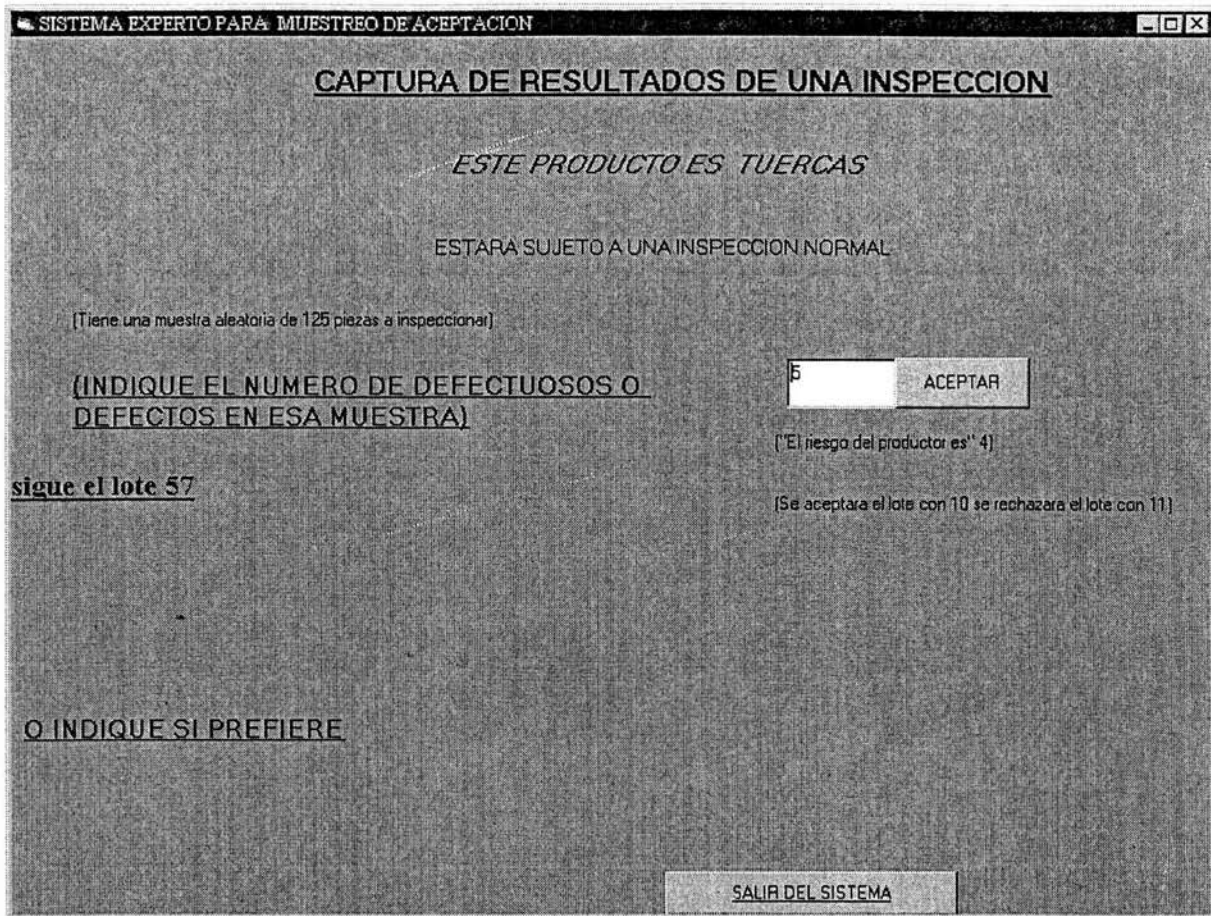


Figura 5.1.11

5.2 EJEMPLO DE LA APLICACIÓN DEL SISTEMA EXPERTO PARA DAR DE ALTA UN PRODUCTO NUEVO.

Se selecciona de la pantalla principal (figura 5.1.1) el botón CONSULTA lo que dará lugar a que aparezca una pantalla de decisión (figura 5.1.8) con dos botones llamados “con historial existente” y “producto nuevo”.

Al seleccionar al botón “producto nuevo” aparece la pantalla que se puede ver en la figura 5.2.1. Para registrar un nuevo producto hay que seleccionar alguno de los botones habilitados para poder ingresar un producto nuevo.

Como lo muestra la figura 5.2.1 en este sistema existe un botón deshabilitado con nombre “TUERCAS” lo que indica que ya está ocupado ese espacio de la base de datos, por un elemento. También se observa que hay dos lugares abiertos para la captura de otro producto (un nuevo producto). Los lugares vacíos tienen por nombre PRODUCTO UNO y PRODUCTO TRES.

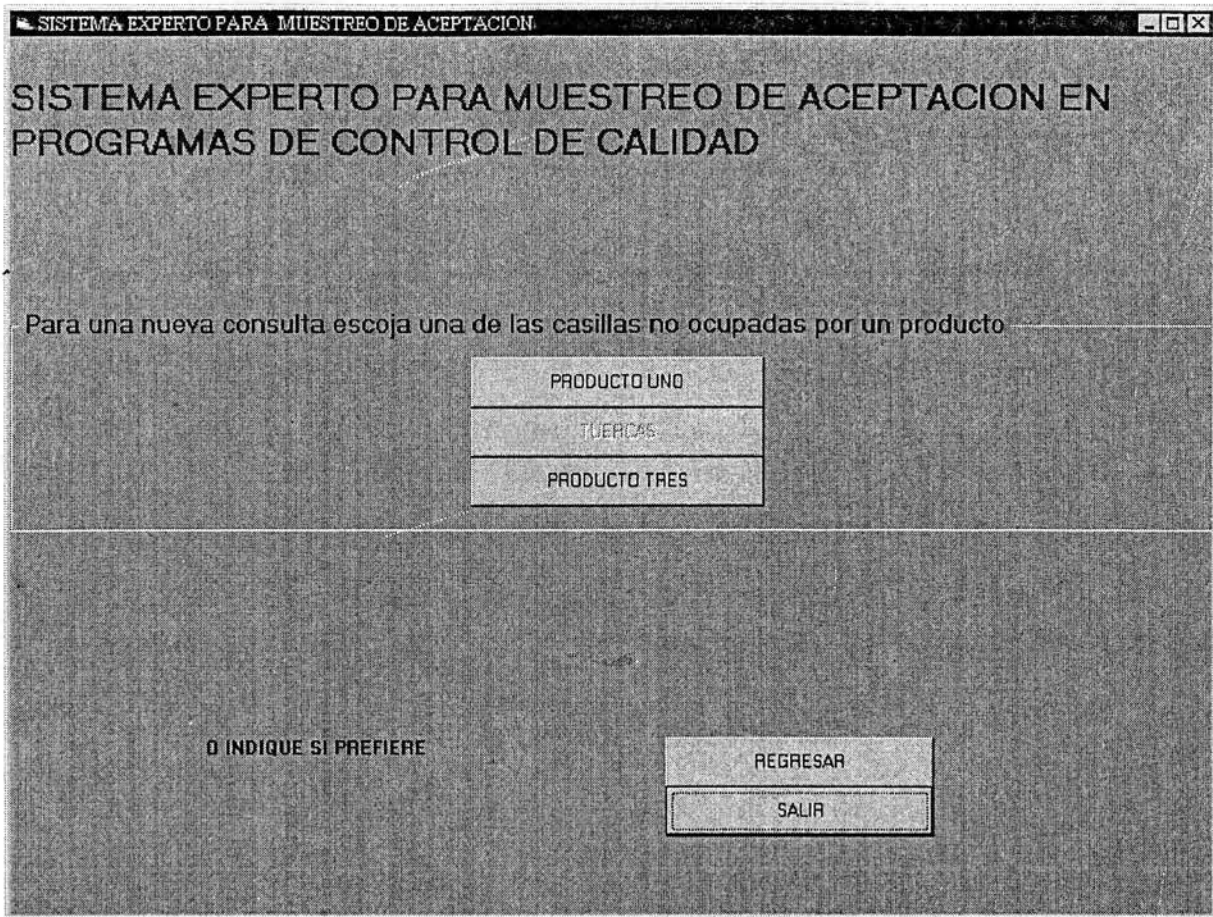


Figura 5.2.1

Dado que el sistema empieza su estudio por un producto nuevo se selecciona en la pantalla que muestra la figura 5.2.1 el botón llamado PRODUCTO UNO para poder introducir el nombre del producto nuevo que lo identificará a lo largo de todo el análisis.

La pantalla siguiente es la preparada para tomar el nombre del producto nuevo en la caja de dialogo que aparece con la leyenda “Nombre del producto a inspeccionar nuevo” como se ilustra en la figura 5.2.2.

En este caso nuestro producto sometido a una inspección se llamará TORNILLOS. Después de introducir el nombre se acepta el mensaje para concluir con la introducción de el nombre del nuevo producto. Si se cancela el mensaje se volverá a la pantalla de la figura 5.2.1. sin que sufra ningún cambio el sistema.

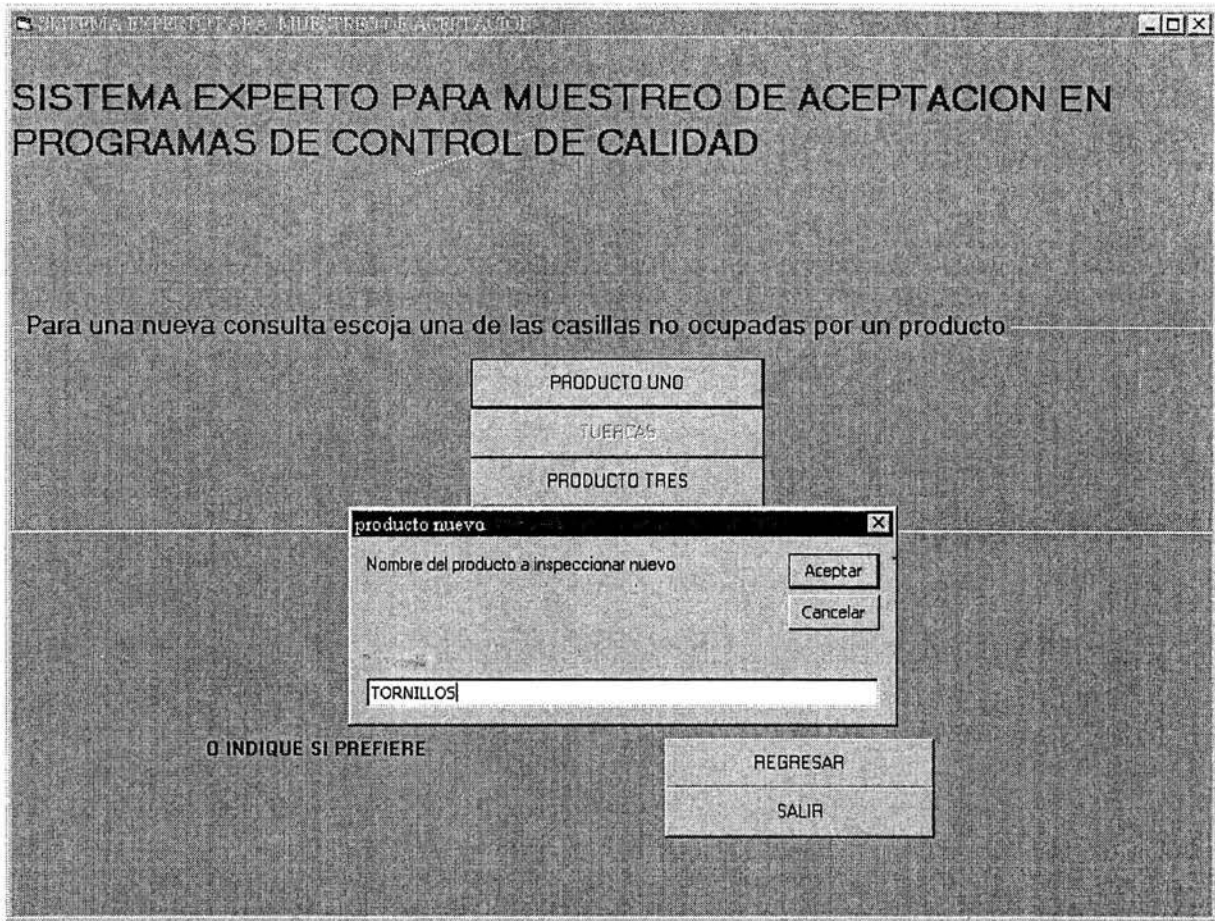


Figura 5.2.2

La pantalla de la figura 5.2.3 aparece después de haber sido aceptado el nombre del producto. Esta pantalla es llamada CAPTURA DE DATOS PARA UN PRODUCTO NUEVO y recoge el tamaño del lote y el riesgo del producto.

Es necesario para comenzar la inspección introducir los datos que caracterizarán a ese producto y que son forzosamente requeridos para poder obtener información de la base de datos del sistema que contiene las especificaciones del estándar militar. Si no se introduce el tamaño del lote o el riesgo del productor AQL el sistema no captura el producto nombrado anteriormente.

El tamaño del lote se introduce en el cuadro de texto que tiene el cursor y posteriormente se acepta para que se prosiga con la selección del riesgo del productor como se observa en la figura 5.2.4

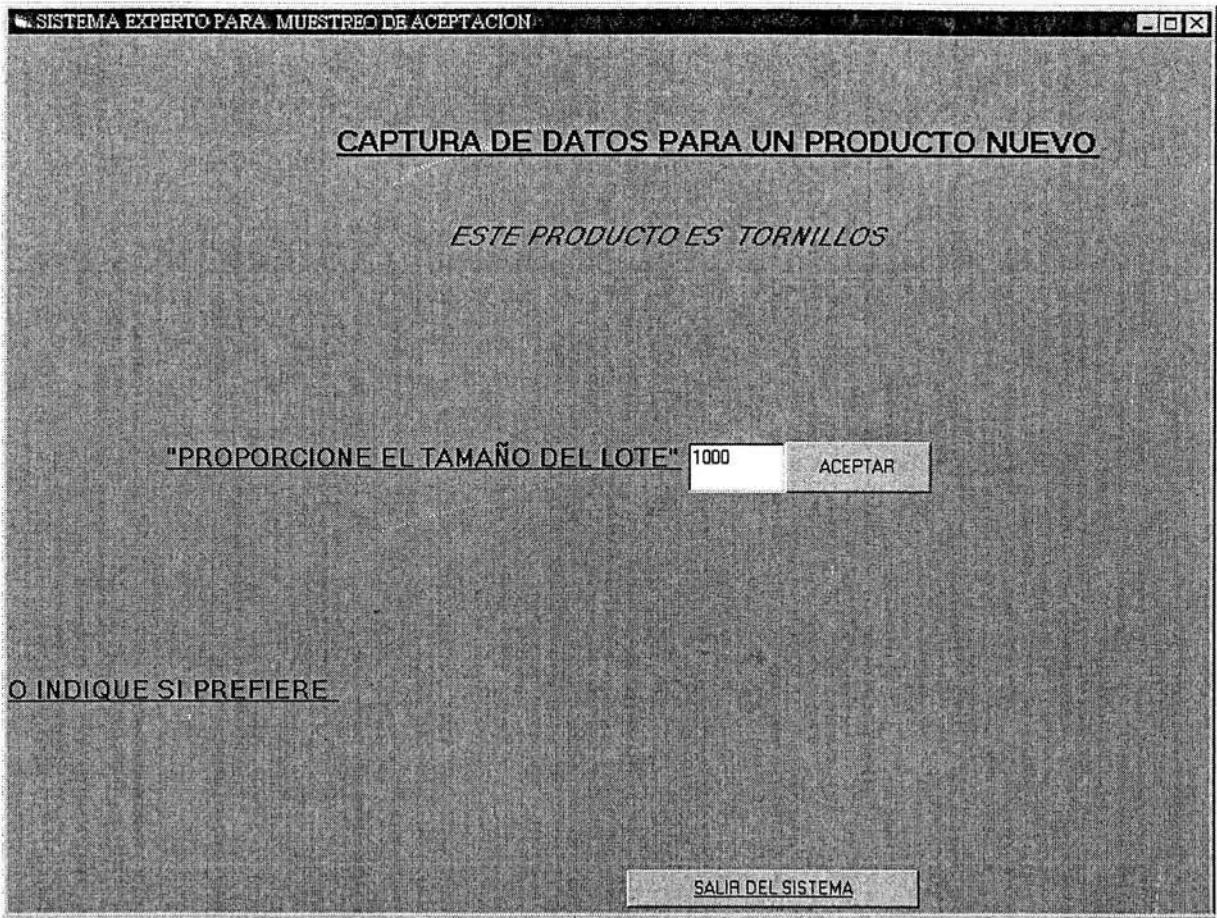


Figura 5.2.3

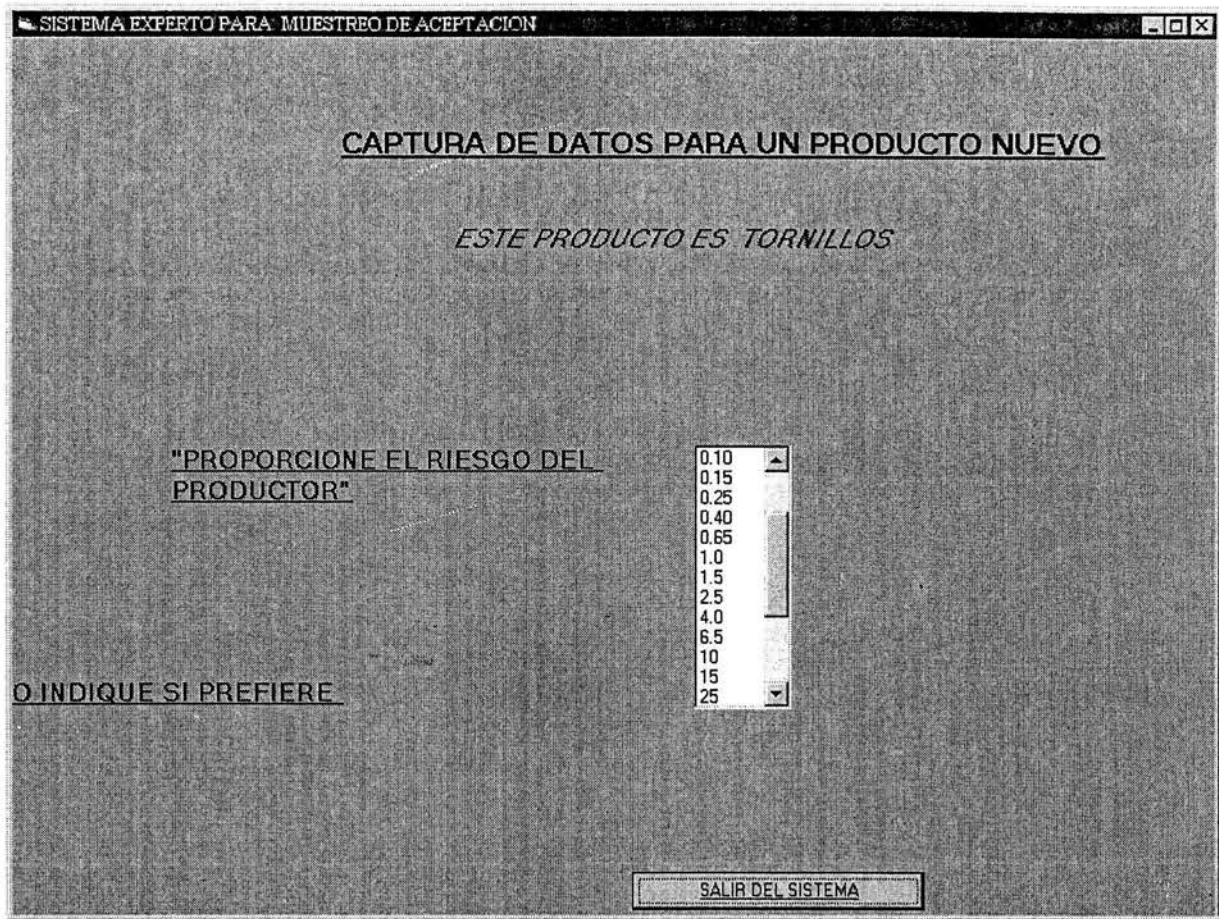


Figura 5.2.4

El riesgo del productor se selecciona de la lista que proporciona el sistema haciendo doble clic sobre el parámetro deseado. Automáticamente se muestra la pantalla que comienza con la captura de los resultados de la inspección, y que siempre se empezará en una inspección normal, como puede verse en la figura 5.2.5.

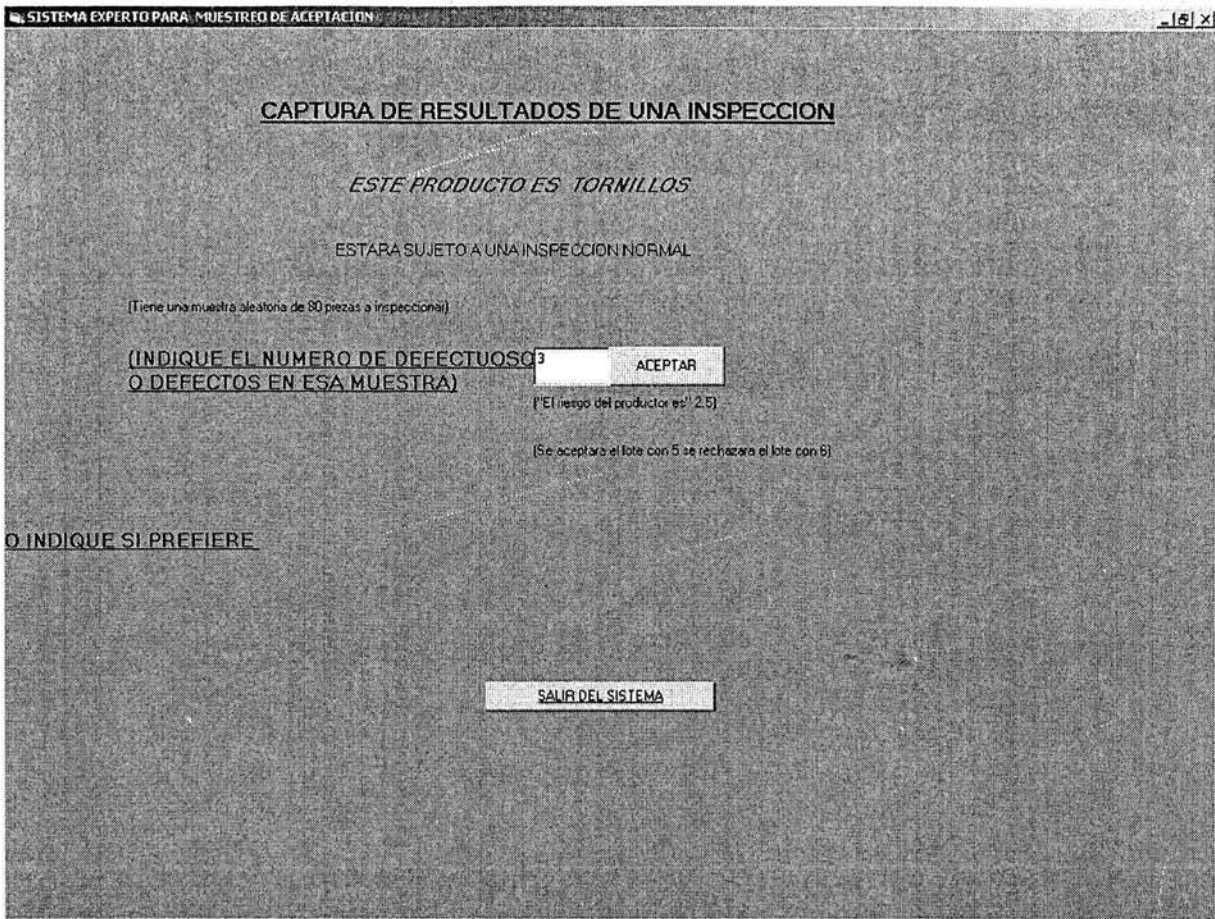


Figura 5.2.5

En la pantalla anterior llamada CAPTURA DE RESULTADOS DE UNA INSPECCION (figura5.2.5) también se da información al usuario acerca del tamaño que deberá tener la muestra, en este caso es de 80 piezas a inspeccionar, al igual que se indica el riesgo del productor (el riesgo del productor es 2.5) y el número de elementos defectuosos con los que se aceptará el lote y el número de elementos defectuosos con los que se rechazará el lote.

Se observa en esta pantalla (figura5.2.5) que se acepta el lote con 5 elementos defectuosos y se rechaza el lote con 6 elementos defectuosos.

Se introduce el número de elementos defectuosos en el cuadro de texto que se encuentra enseguida del mensaje "INDIQUE EL NUMERO DE DEFECTUOSOS O DEFECTOS EN ESA MUESTRA" para después acreditar el dato proporcionado con el botón de aceptar. Con ésta acción aparecerá la pantalla llamada "ACEPTACIÓN O RECHAZO DEL LOTE" que ilustra la figura 5.2.6.

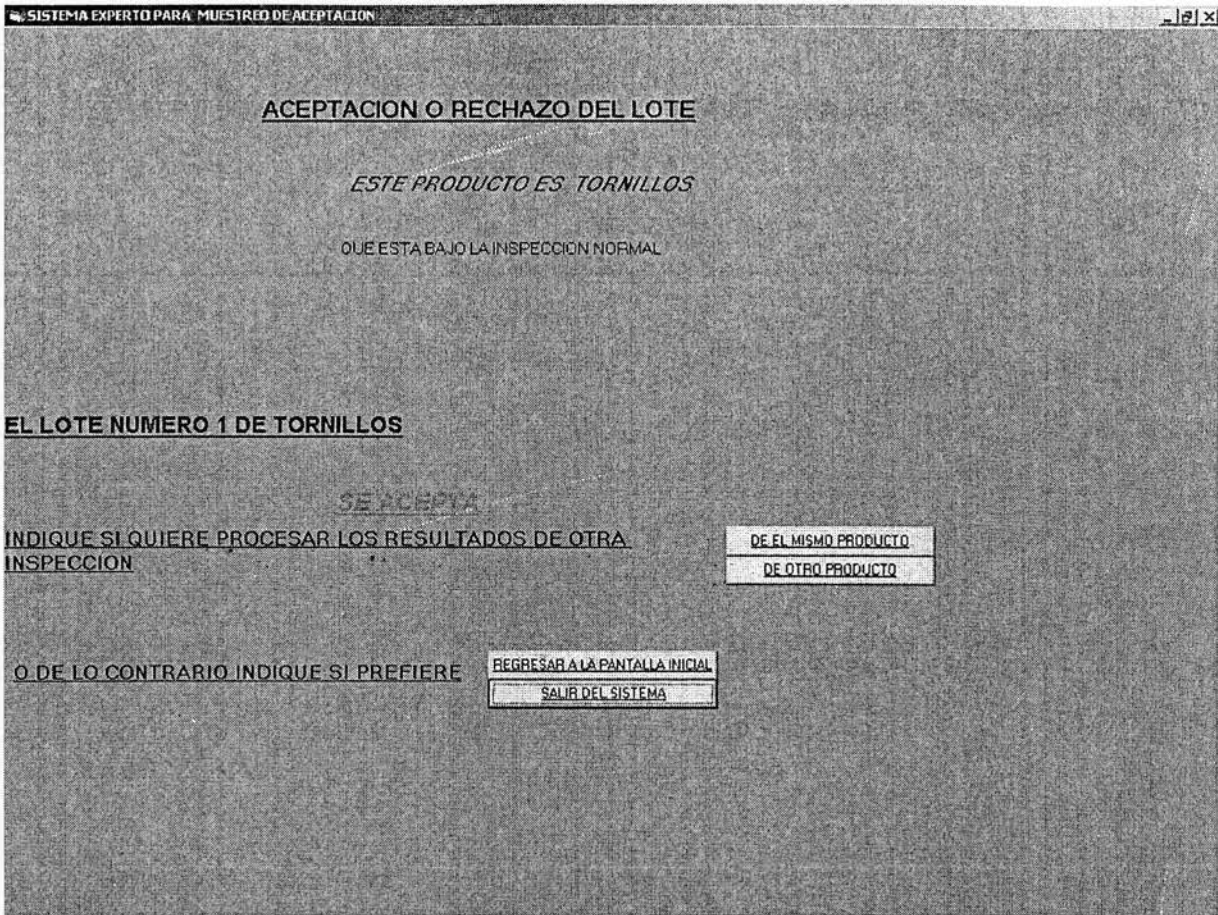


Figura 5.2.6

Esta pantalla le da información al usuario sobre la inspección aplicada al producto TORNILLOS. Como se ve el mensaje “EL LOTE NUMERO 1 DE TRONILLOS” indica que número de lote que fue inspeccionado y el mensaje “SE ACEPTA” indica que ese lote es aceptado por el estándar militar.

Se selecciona el botón “DE EL MISMO PRODUCTO“ para volver a aplicar otra inspección al siguiente lote de ese mismo producto, en este caso el producto TORNILLOS.

Se selecciona el botón “DE OTRO PRODUCTO” para poder cambiar el producto inspeccionado, es decir para poder elegir otro producto del sistema. Esta acción regresa a la pantalla mostrada en la figura 5.1.8 del tema 5.1 de este capítulo 5.

Se selecciona el botón “REGRESAR A LA PANTALLA INICIAL” para volver a la pantalla inicial mostrada en la figura 5.1.1 del tema 5.1 de este capítulo 5.

La figura 5.2.7 muestra la pantalla obtenida al seleccionar el botón “DE EL MISMO PRODUCTO“. Como se observa la inspección se aplica al lote dos del producto TORNILLOS. La figura 5.2.8 muestra la pantalla de aceptación o rechazo del lote. Como se puede observar el lote dos del producto TORNILLOS es rechazado.

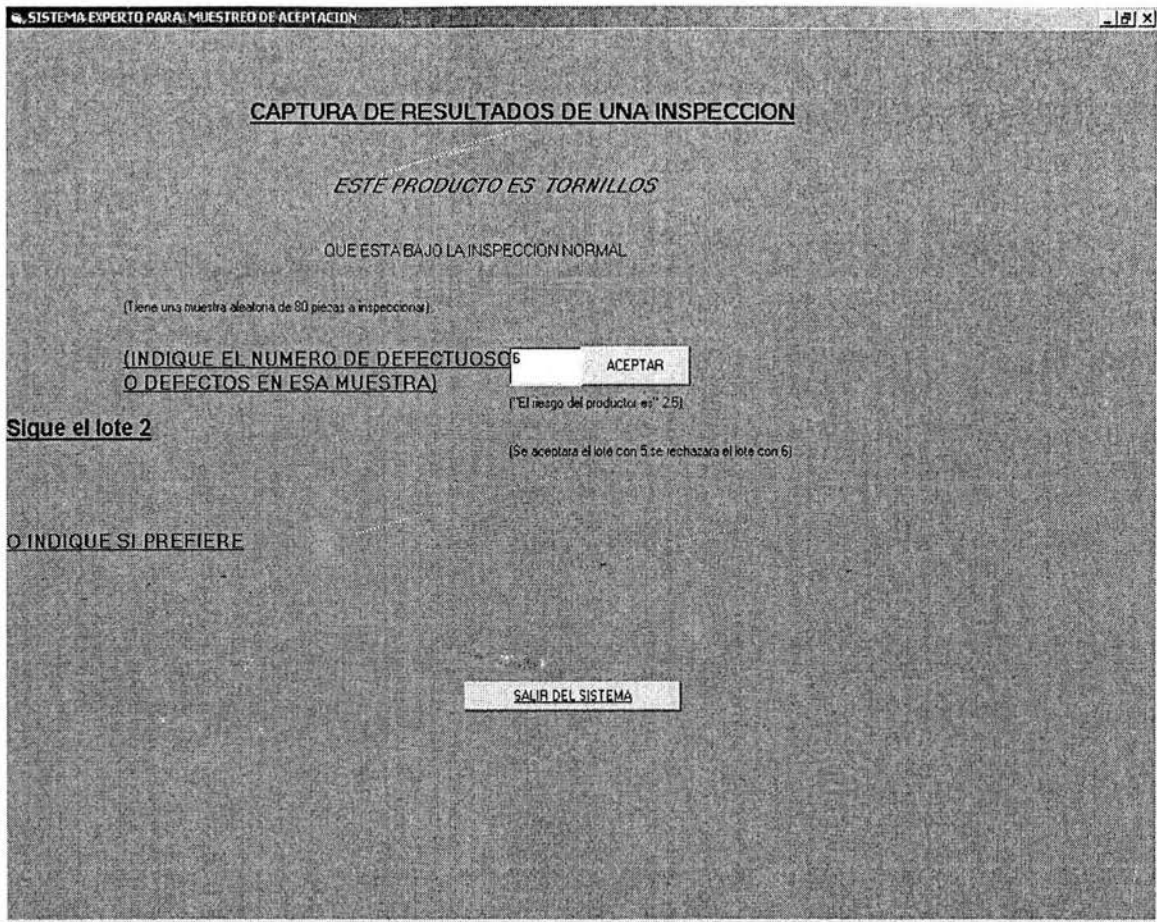


Figura 5.2.7

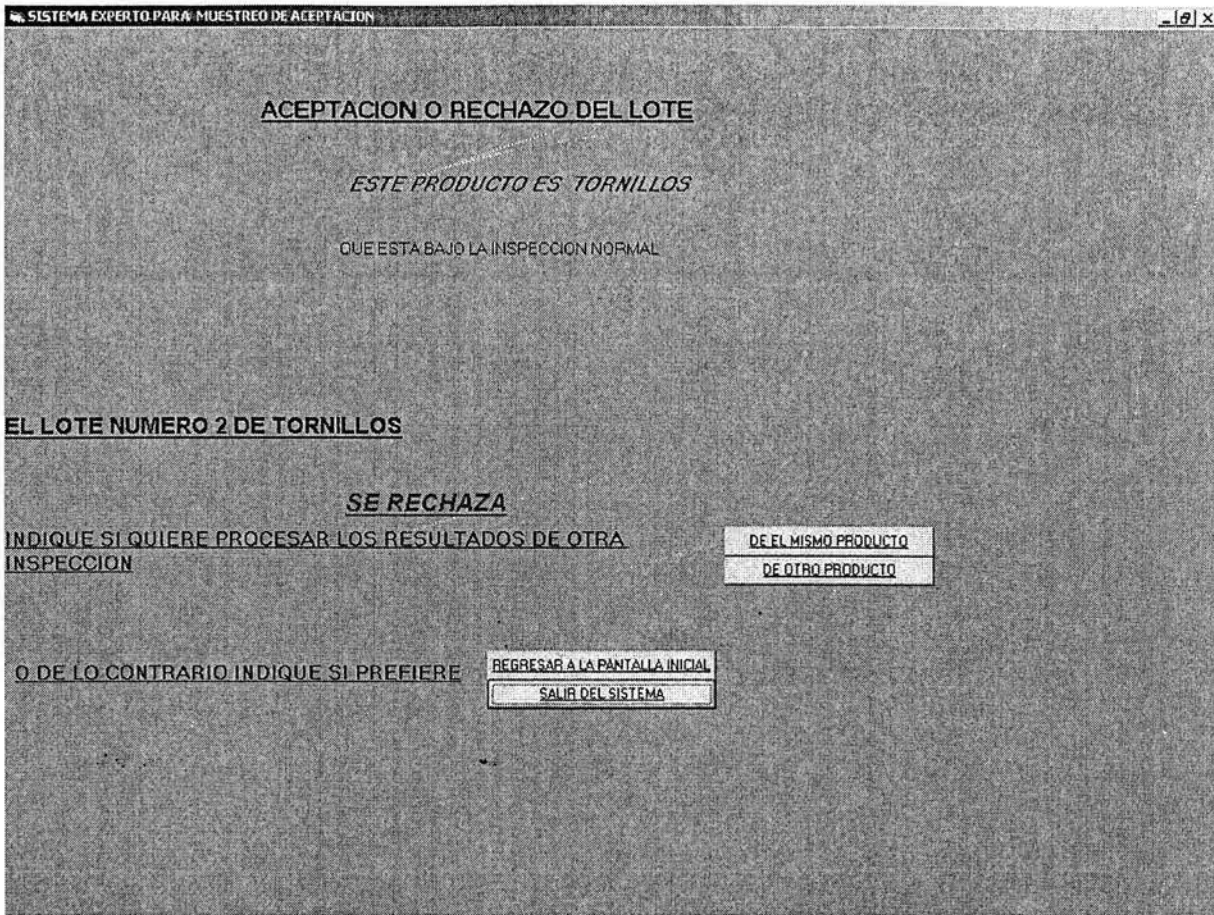


Figura 5.2.8

5.3 EJEMPLO DE LA APLICACIÓN DEL SISTEMA EXPERTO EN EL CAMBIO DE INSPECCIÓN NORMAL A REDUCIDA EN UN PRODUCTO EXISTENTE.

Para que se acredite el cambio de una inspección normal a un inspección reducida es necesario que se cumplan la regla que confirman ese cambio:

- Cuando el sistema obtiene “los últimos consecutivos diez lotes aceptados”,
- y cuando “el número total de defectos sea menor que el número limite marcado en las tablas correspondientes al estándar militar que fue la línea de desarrollo”
- y cuando el sistema obtiene una afirmación a la pregunta “¿el sistema es estable?”
- y cuando el sistema obtiene una afirmación a la pregunta “¿se autoriza el cambio por una persona capacitada?”.

En la figura 5.3.1 se muestra el historial del producto TORNILLOS el cual indica que se han inspeccionado 12 lotes y que los últimos 10 han sido aceptados, pero el número total de defectos encontrados en esos últimos 10 lotes no es menor al señalado en el estándar militar por lo que no es propicio cambiar a la inspección reducida.

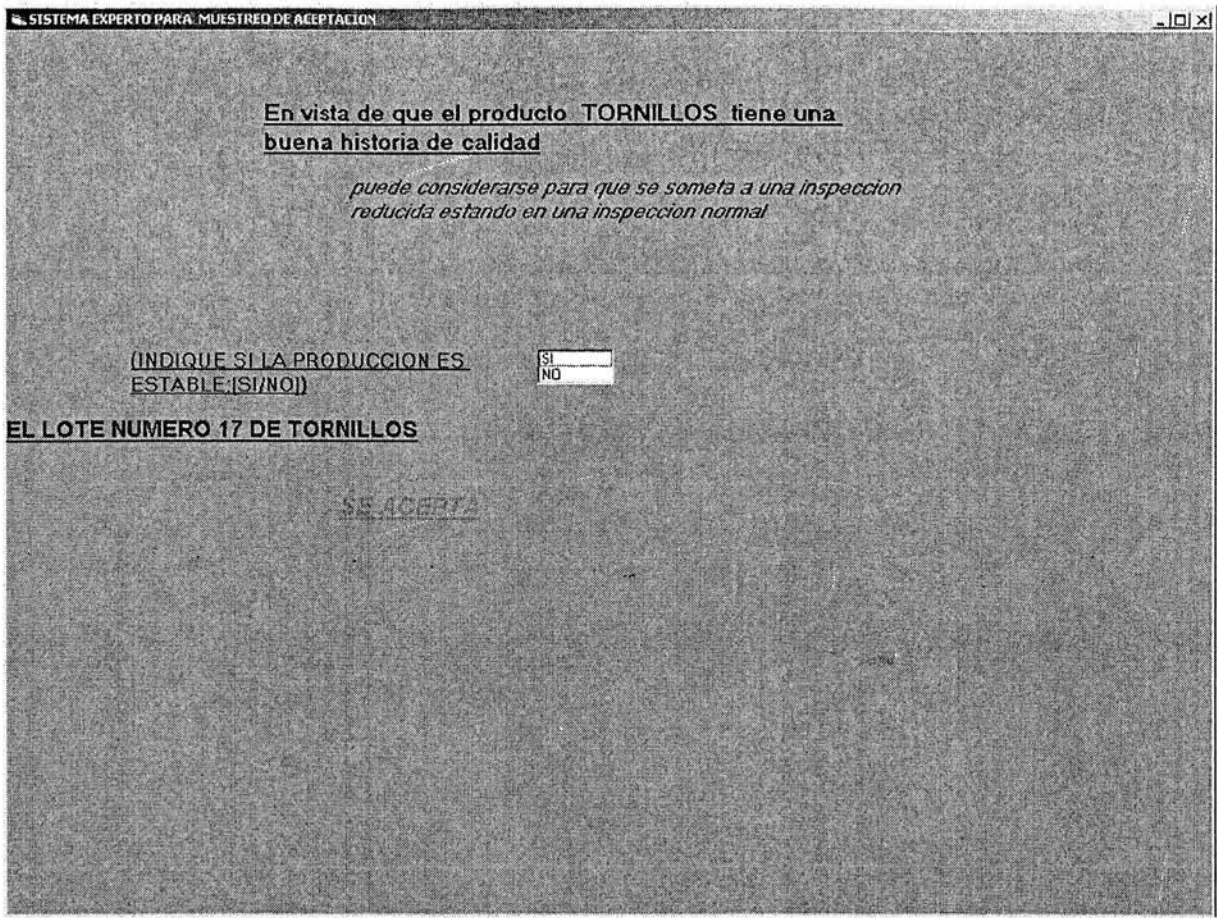


Figura 5.3.2

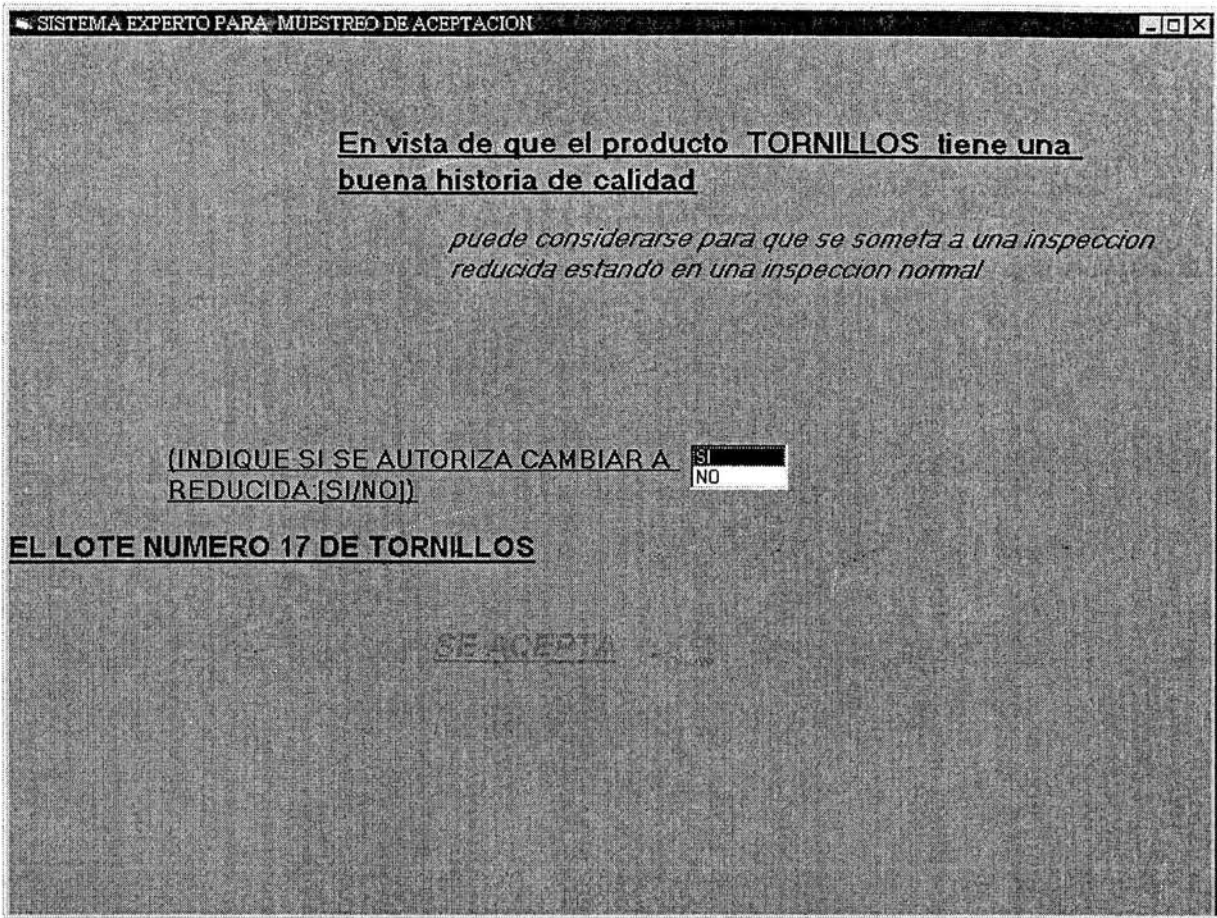


Figura 5.3.3

Al contestar afirmativamente a las dos preguntas antes descritas el sistema está listo para cambiar a la inspección reducida como se puede observar en la figura 5.3.4

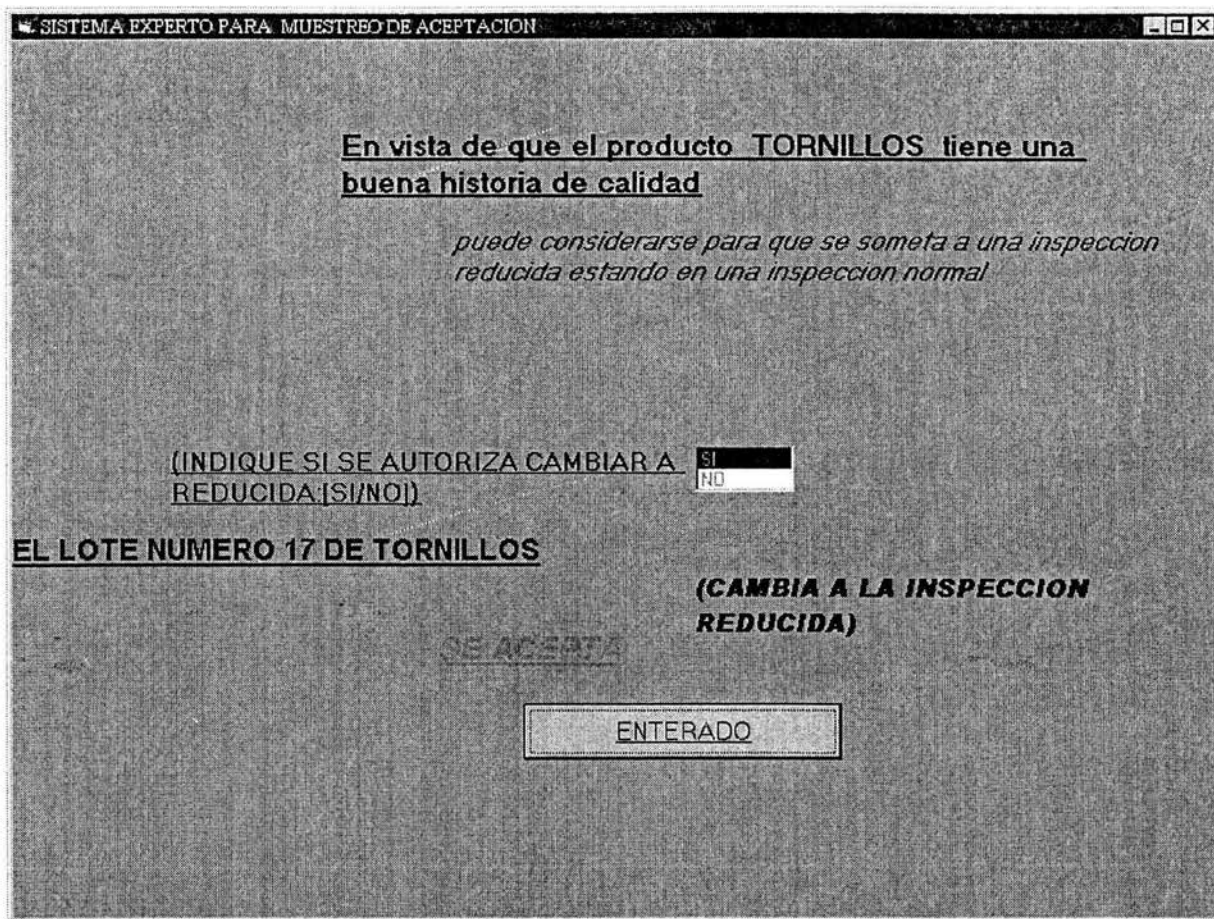


Figura 5.3.4

Se selecciona el botón ENTERADO para confirmar el cambio de inspección. Una vez aseverado este cambio el sistema experto pasará de la inspección normal a la inspección reducida.

La pantalla siguiente ilustra que el sistema ha cambiado a la inspección reducida, esto se le indica al usuario en el mensaje “ESTARA SUJETO A UNA INSPECCION REDUCIDA”. En la figura 5.3.5 se observa lo anterior.

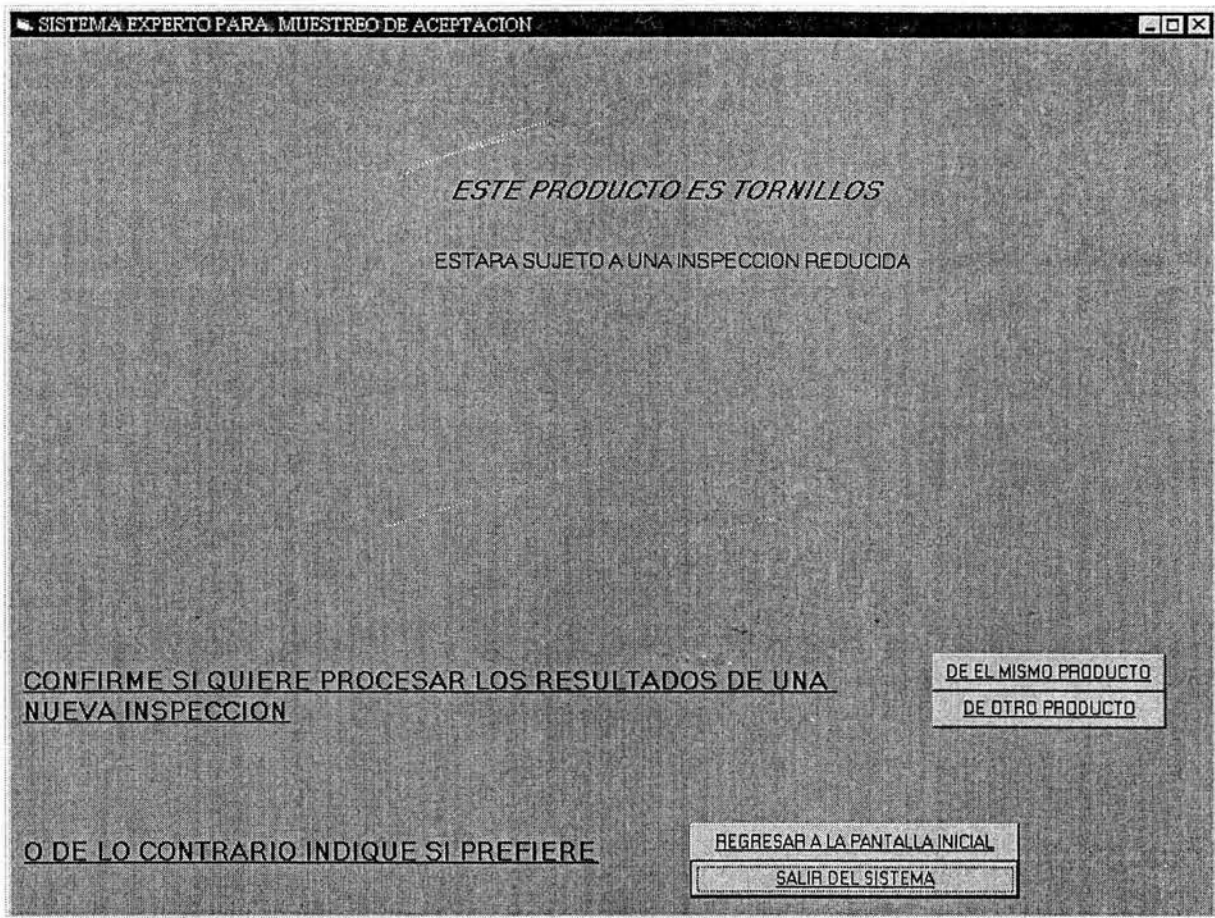


Figura 5.3.5

Si se selecciona el botón “DE EL MISMO PRODUCTO” aparecerá la pantalla anteriormente descrita en la figura 5.2.7 llamada CAPTURA DE RESULTADOS DE UNA INSPECCION que ahora ya no se encontrará referida a una inspección normal, si no que hará referencia a la inspección siguiente (inspección reducida). La figura 5.3.6 ilustra lo anterior.

El usuario también puede dejar pendiente esta inspección del producto TORNILLOS y continuar con otra inspección de otro producto seleccionando el botón “DE OTRO PRODUCTO” que regresara al usuario a la pantalla mostrada en la figura 5.1.8 del tema 5.1 de este capítulo 5.

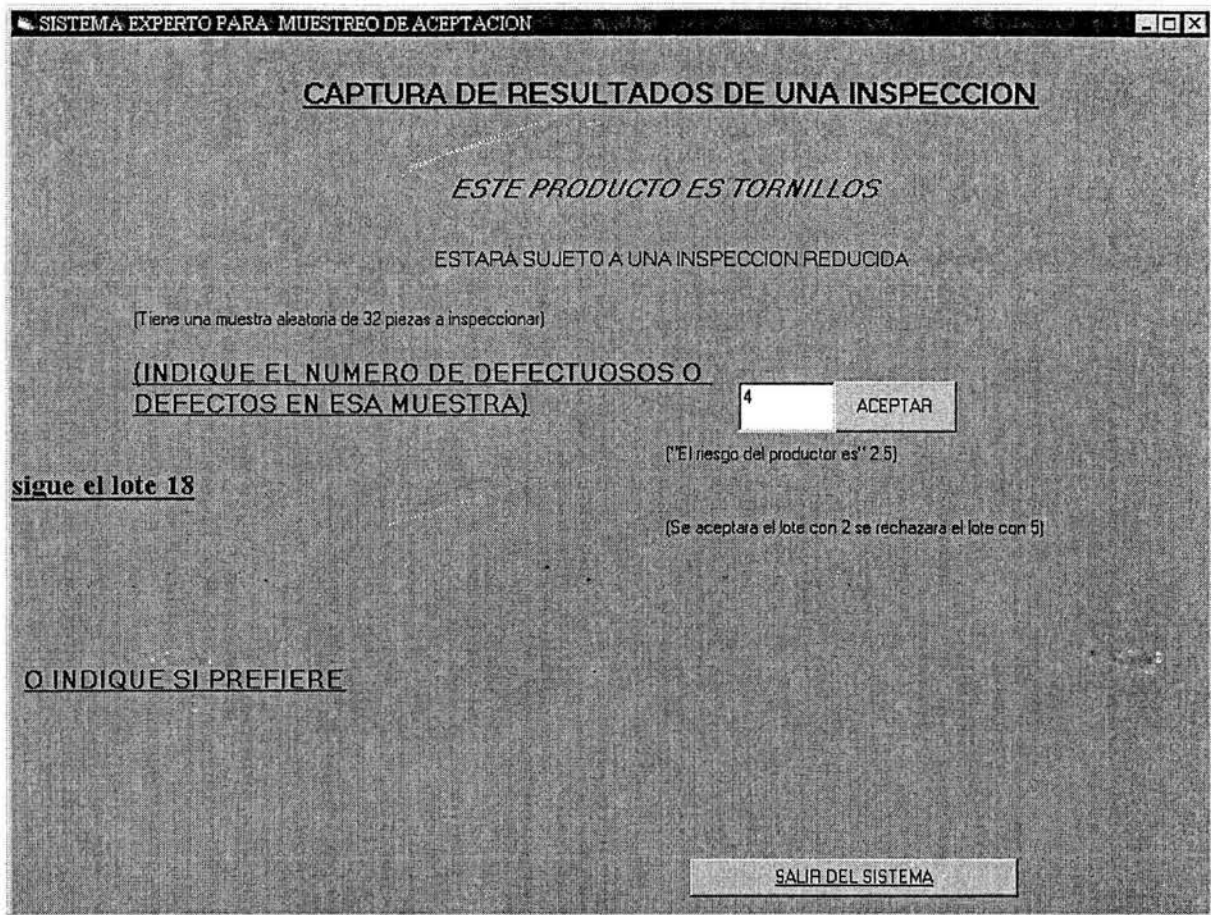


Figura 5.3.6

5.4 EJEMPLO DE LA APLICACIÓN DEL SISTEMA EXPERTO EN EL CAMBIO DE INSPECCIÓN REDUCIDA A NORMAL.

Ahora el producto TORNILLOS ya está en una inspección reducida y los elementos defectuosos se manipularan de acuerdo a las especificaciones del estándar para la inspección reducida. Al aceptar en el botón de ACEPTAR se procesan los datos adquiridos y el sistema indica que el lote 18 es aceptado pero que la inspección debe cambiar nuevamente a una inspección normal como se ilustra en la figura 5.4.1.

Este cambio de inspección reducida a normal se da debido a que la regla indica que un producto estando en una inspección reducida debe regresar a una inspección normal si

- el lote es rechazado
- o el lote es aceptado pero el número de defectos se encuentra entre el número de defectos de aceptación del lote y el número de defectos de rechazo del lote, debido a que 4 defectos están entre 2 y 5 como se mostró en la figura 5.3.6
- o la producción es irregular
- o lo aconsejan otras condiciones.

En este caso el lote se aceptó pero el número de defectos de ese lote esta dentro del intervalo marcado por el número límite de aceptación y el número límite de rechazo. Debido a esto se regresó a la inspección normal.

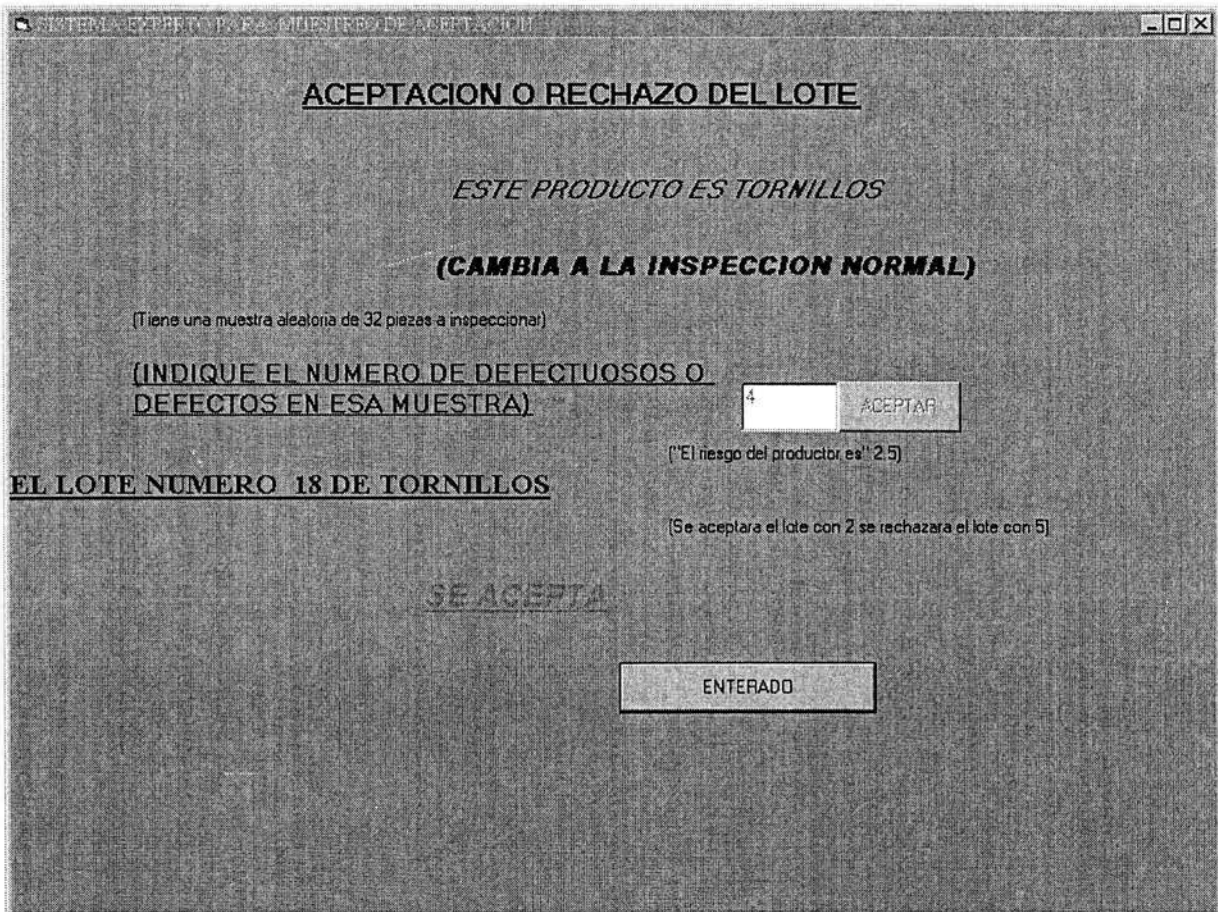


Figura 5.4.1

Para continuar la inspección se acepta el botón de ENTERADO que muestra la pantalla que le indica al usuario que el sistema ha regresado a la inspección normal. El mensaje ESTARA SUJETO A UNA INSPECCION NORMAL afirma este cambio. La figura 5.4.2 ilustra lo anterior.

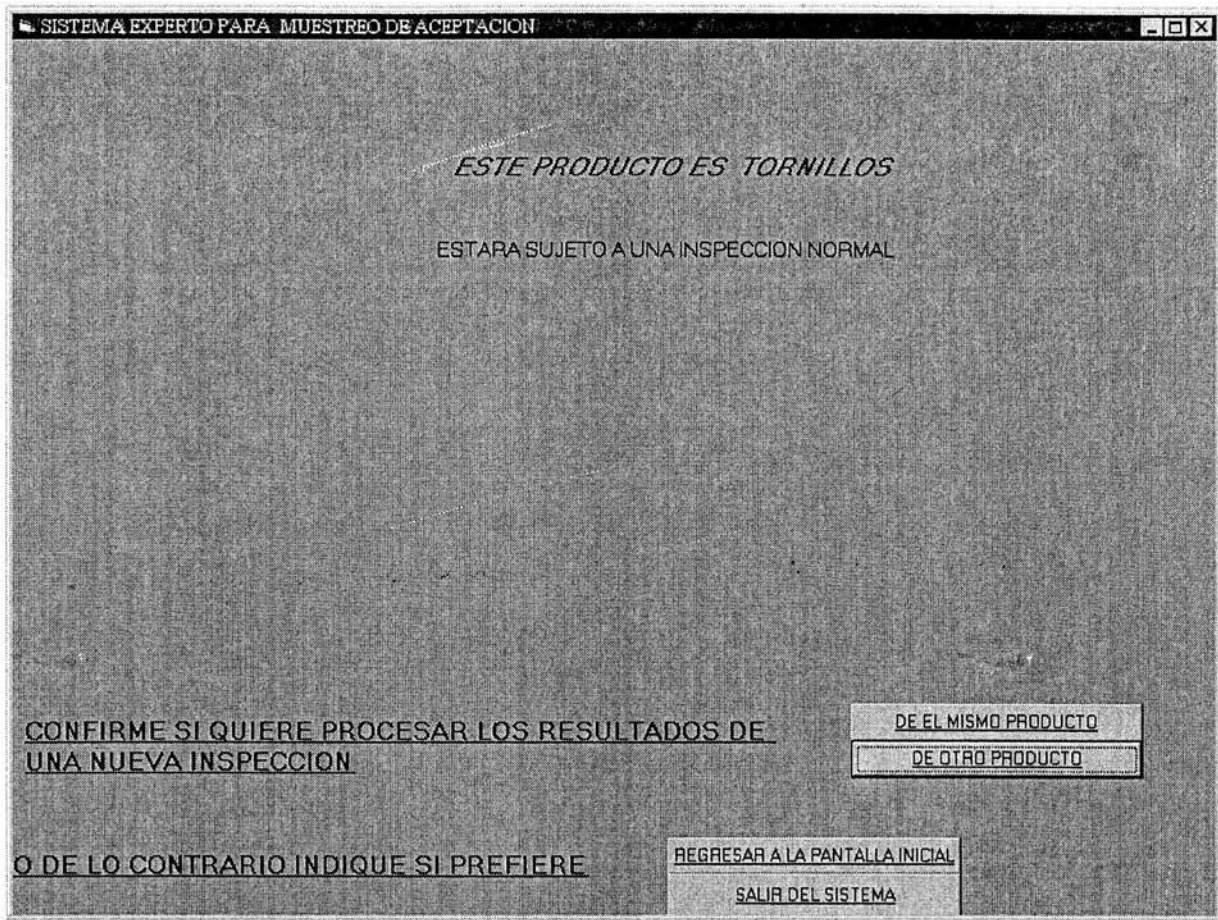


Figura 5.4.2

Si se quiere confirmar que se ha aplicado correctamente la regla de cambio de normal a reducida y que se ha trabajado un lote en esta inspección reducida para regresar a la inspección normal, se debe observar el historial del producto inspeccionado que aparecerá al seleccionar el botón "REGRESAR A LA PANTALLA INICIAL" y posteriormente el botón llamado HISTORIA para después elegir la historia del producto, en este caso TORNILLOS.

El historial del producto TORNILLOS es ilustrado en la figura 5.4.3 que muestra una pantalla donde se observa que la inspección fue propicia para el cambio de normal a reducida y que el sistema aplicó este cambio a la inspección y se trabajó un lote en esta inspección.

En el caso que nos ocupa, dado que el número de defectos, que fue de 4 estuvo entre 2 y 5, el siguiente lote estará sujeto a inspección normal, como se mostró en la figura 5.4.2.

CAPITULO 6

MANTENIMIENTO

Por mantenimiento de un software se entiende que son los cambios que deben hacerse al sistema después de haber sido enviados al usuario para mantener operativo el software que ya fue aceptado y que esta en uso.

El mantenimiento cubre la vida de un software desde el momento en que se instala hasta que deja de ser útil. Con el mantenimiento se modifica un software después de haber sido entregado para corregir fallos, mejorar el rendimiento u otros atributos, o para adaptarlo a un ambiente cambiante. El objetivo es modificar el software existente conservando su integridad.

El comentario común es que el mantenimiento es una actividad “después de”. Ocurre después de que el producto esté en operación.

Existen cuatro categorías de mantenimiento que se definen a continuación:

- **Correctivo:** cambios necesarios por errores. El problema es localizar la especificación original para determinar para que fue diseñado originalmente el sistema.
- **Adaptativo:** cualquier esfuerzo iniciado como resultado de cambios en el entorno en el que opera el sistema software. El usuario no ve un cambio directo en la operación del sistema, pero el gestor de mantenimiento debe aplicar recursos para realizar el cambio.
- **Perfectivo:** todos los cambios, inserciones, borrados, modificaciones, extensiones y mejoras hechas al sistema para complacer la evolución o expansión de las necesidades del usuario. Todos los nuevos requisitos del usuario entran en esta categoría.
- **Preventivo:** mantenimiento realizado con el propósito de prevenir problemas antes de que ocurran.

El software realizado en esta tesis se considera aceptado y funcional como un programa de demostración que puede ser ampliado posteriormente si se quiere implantar como un sistema empresarial. Debido a esto el mantenimiento correctivo no es necesario dado que el sistema ya fue probado y no proyecta ningún error en su operación.

Para este caso se tendría que recurrir al mantenimiento adaptativo y perfectivo. Es decir se tendrían que considerar en el sistema más bases de datos para un mayor número de productos o para el número de productos que sean requeridos por alguna empresa y que puedan facilitar al usuario el trabajo, ya que en este caso el sistema solo cuenta con tres lugares habilitados. Esto nos llevaría a la modificación de la interfase de usuario agrandando las opciones de elección de productos y como es claro la ampliación del tamaño en cuanto a bases de datos.

En cuanto al mantenimiento preventivo este tiene lugar en el sistema si el estándar que es la base de desarrollo del sistema experto de muestreo para aceptación en programas de control de calidad se actualizara. Como consecuencia se tendrían que actualizar los datos necesarios para que el programa funcione debidamente, modificando también la base de conocimiento del sistema desarrollado.

CAPITULO 7

CONCLUSIONES

- El software desarrollado en esta tesis es una aplicación de los sistemas expertos que da tratamiento a un problema particular basado en la aplicación de acciones repetitivas.
- El desarrollo del sistema experto se llevó a cabo en base a un cascarón de libre distribución llamado Clips, manejando un encadenamiento hacia adelante para éste problema.
- Ese cascarón quedó integrado con Visual Basic, lo que permite que la interfase con el usuario se clara y amigable, dado que Visual Basic maneja todos los elementos de la interfase de Windows.
- La forma de integrar Clips y Visual Basic se realiza a través de Clips OCX que se encarga de transmitir la información entre Clips con la interfase de Visual Basic.
- La base de datos del sistema se realizó en Excel y Access. Dado que en Excel tiene la ventaja de utilizar comandos como el de cortar y pegar información es más fácil capturar bases de datos de gran tamaño sin cometer errores, como se hizo con las bases de datos del estándar militar MIL-STD-105D que contienen muchos datos. Por esta razón la utilización de Excel para realizar estas bases de datos fue muy apropiada. Por otra parte, Access se utilizó para la realización de las bases de datos que el sistema experto necesita para llevar un control de el número de lotes inspeccionados y los defectos encontrados en dichos lotes, información que se introduce en la base de datos de Access ya estando en ejecución el sistema experto.
- Visual Basic permite integrar estas dos herramientas a su entorno de desarrollo para manipular, sacar o guardar información en la base de datos de Access y Excel a través de programación en la interfase gráfica de Visual Basic.
- La aplicación del problema muestreo para aceptación en un sistema experto es atinada porque el conocimiento se puede representar en base a reglas y porque la aplicación de la norma del estándar militar MIL-STD-105D, por ser repetitiva, puede volverse tediosa y resultar en errores, lo cual se evita con un sistema experto.
- Para propósito de esta tesis se limitó a tres el número de artículos sujetos a inspección en la base de datos. Debe destacarse que el haber usado solo tres artículos no es limitante del enfoque utilizado, y que la misma arquitectura permite extender el número de artículos al necesario para reflejar situaciones de la vida real. Lo anterior permite ejemplificar la sencillez de dar de alta o de baja a los artículos de la base de datos como parte integral del sistema.
- El sistema experto de esta tesis se realizó con software de libre distribución como fueron las herramientas de Clips, Visual Basic. Se muestra que estas herramientas pueden ser

utilizadas adecuadamente para resolver el problema de muestreo para aceptación, no solo contemplando la norma del estándar militar MIL-STD-105D sino extendiéndose a otros planes de muestreo por atributos. Entre estos se puede destacar los planes de muestreo simple, múltiple y secuencial. La ventaja principal de estos planes con respecto al muestreo simple es que se puede reducir la cantidad total de inspección requerida. La desventaja de estos planes, en general, es que los procedimientos son más complejos, lo que puede ocasionar errores de aplicación. Con un sistema experto se eliminan los riesgos de cometer dichos errores, lo que permite capitalizar una de las principales ventajas de los sistemas expertos.

BIBLIOGRAFÍA

- (1) H.C.CHARBONNEAU, G.L.WEBSTER, CONTROL DE CALIDAD, McGRAW-HILL, MÉXICO, 1989.
- (2) ARCHESON J.DUNCAN, CONTROL DE CALIDAD Y ESTADÍSTICA INDUSTRIAL, ALFAOMEGA, MÉXICO D.F, 1996.
- (3) JOSEPH JIARRATANO, GARY RILEY, SISTEMAS EXPERTOS PRINCIPIOS Y PROGRAMACIÓN , INTERNATIONAL THOMSON EDITORES, MÉXICO, 1998.
- (4) <http://shannon.unileon.es/~diecbc/asignaturas/IngenieriaDelConocimiento/practicas/>
- (5) http://www.ingenieroseninformatica.org/recursos/tutoriales/sist_exp/cap1.php
- (6) <http://www.iespana.es/uccsistemaexperto/>
- (7) <http://html.rincondelvago.com/sistema-experto.html>
- (8) <http://ourworld.compuserve.com/homepages/marktoml/clipstuf.htm>
- (9) <http://personal.lig.bellsouth.net/t/e/teolson/OCXInstallPage.htm>
- (10) http://www.anayamultimedia.es/servicios/libros/conozca_win98.pdf
- (11) <http://www.amauta.edu.mx/Microsoft%20Access/Microsoft%20Access.htm>

TABLA I Letras codificadas del tamaño de la muestra

Medida del lote o conjunto			Niveles de inspección especial				Niveles de inspección general		
			S-1	S-2	S-3	S-4	I	II	III
2	a	8	A	A	A	A	A	A	B
9	a	15	A	A	A	A	A	B	C
16	a	25	A	A	B	B	B	C	D
26	a	50	A	B	B	C	C	D	E
51	a	90	B	B	C	C	C	E	F
91	a	150	B	B	C	D	D	F	G
151	a	280	B	C	D	E	E	G	H
281	a	500	B	C	D	E	F	H	J
501	a	1 200	C	C	E	F	G	J	K
1 201	a	3 200	C	D	E	G	H	K	L
3 201	a	10 000	C	D	F	G	J	L	M
10 001	a	35 000	C	D	F	H	K	M	N
35 001	a	150 000	D	E	G	J	L	N	P
150 001	a	500 000	D	E	G	J	M	P	Q
500 001	y	mayores	D	E	H	K	N	Q	R

TABLA IV-B Planes de muestre único le para inspección rigurosa.(Tabla muestra)

Letra codificada de la medida de la muestra	Medida de la muestra	Niveles de calidad aceptable (inspección rigurosa)																											
		0.010	0.015	0.025	0.040	0.065	0.10	0.15	0.25	0.40	0.65	1.0	1.5	2.5	4.0	6.5	10	15	25	40	65	100	150	250	400	650	1 000		
		Ac Re	Ac Re	Ac Re	Ac Re	Ac Re	Ac Re	Ac Re	Ac Re	Ac Re	Ac Re	Ac Re	Ac Re	Ac Re	Ac Re	Ac Re	Ac Re	Ac Re	Ac Re	Ac Re	Ac Re	Ac Re	Ac Re	Ac Re	Ac Re	Ac Re	Ac Re	Ac Re	
A	2	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓		
B	3	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓		
C	5	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓		
D	8	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓		
E	13	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓		
F	20	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓		
G	32	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓		
H	50	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓		
J	80	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓		
K	125	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓		
L	200	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓		
M	315	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓		
N	500	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓		
P	800	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓		
Q	1 250	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓		
R	2 000	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓		
S	3 150	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓		

= Usese primero el plan de muestreo indicado por la flecha hacia abajo. Si la medida de la muestra iguala o excede la medida del lote o conjunto, hágase una inspección al 100%.
 = Usese primero el plan de muestreo indicado por la flecha hacia arriba.
 Ac = Número de aceptación.
 Re = Número de rechazo.

TABLA II-C Planes de muestreo único para inspección reducida (Tabla maestra)

Letra codificada de la medida de la muestra	Medida de la muestra	Niveles de calidad aceptable (inspección reducida) †																											
		0.010	0.015	0.025	0.040	0.065	0.10	0.15	0.25	0.40	0.65	1.0	1.5	2.5	4.0	6.5	10	15	25	40	65	100	150	250	400	650	1000		
		Ac Re	Ac Re	Ac Re	Ac Re	Ac Re	Ac Re	Ac Re	Ac Re	Ac Re	Ac Re	Ac Re	Ac Re	Ac Re	Ac Re	Ac Re	Ac Re	Ac Re	Ac Re	Ac Re	Ac Re	Ac Re	Ac Re	Ac Re	Ac Re	Ac Re	Ac Re	Ac Re	
A	2	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓		
B	2	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓		
C	2	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓		
D	3	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓		
E	5	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓		
F	8	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓		
G	13	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓		
H	20	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓		
J	32	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓		
K	50	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓		
L	80	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓		
M	125	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓		
N	200	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓		
P	315	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓		
Q	500	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓		
R	800	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓		

108



 = Usese primero el plan de muestreo indicado por la flecha hacia abajo. Si la muestra iguala o excede la medida del lote o conjunto, hágase una inspección al 100%.
 = Usese primero el plan de muestreo indicado por la flecha hacia arriba.
Ac = Número de aceptación
Re = Número de rechazo
† = Si se ha excedido el número de aceptación, pero no se ha alcanzado el número de rechazo, se acepta el lote, pero se reinstala la inspección normal (Véase 10.1.4)

TABLA 10.10

Números límite para inspección abreviada (Mil. Std. 105D, tabla VIII)

Números de unidades de muestra de los últimos 10 lotes o conjuntos	Nivel aceptable de calidad																									
	0.010	0.015	0.025	0.040	0.065	0.10	0.15	0.25	0.40	0.65	1.0	1.5	2.5	4.0	6.5	10	15	25	40	65	100	150	250	400	650	1000
20 - 29	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	0	0	2	4	8	14	22	40	68	115	181
30 - 49	*	*	*	*	*	*	*	*	*	*	*	*	*	*	0	0	1	3	7	13	22	36	63	105	178	277
50 - 79	*	*	*	*	*	*	*	*	*	*	*	*	*	0	0	2	3	7	14	25	40	63	110	181	301	
80 - 129	*	*	*	*	*	*	*	*	*	*	*	0	0	2	4	7	14	24	42	68	105	181	297			
130 - 199	*	*	*	*	*	*	*	*	*	*	0	0	2	4	7	13	25	42	72	115	177	301	490			
200 - 319	*	*	*	*	*	*	*	*	*	0	0	2	4	8	14	22	40	68	115	181	277	471				
320 - 499	*	*	*	*	*	*	*	*	0	0	1	4	8	14	24	39	68	113	189							
500 - 799	*	*	*	*	*	*	*	0	0	2	3	7	14	25	40	63	110	181								
800 - 1249	*	*	*	*	*	*	0	0	2	4	7	14	24	42	68	105	161									
1250 - 1999	*	*	*	*	*	0	0	2	4	7	13	24	40	69	110	169										
2000 - 3149	*	*	*	*	*	0	0	2	4	8	14	22	40	68	115	181										
3150 - 4999	*	*	*	*	0	0	1	4	8	14	24	38	67	111	186											
5000 - 7999	*	*	*	0	0	2	3	7	14	25	40	63	110	181												
8000 - 12499	*	*	0	0	2	4	7	14	24	42	68	105	181													
12500 - 19999	*	0	0	2	4	7	13	24	40	69	110	169														
20000 - 31499	0	0	2	4	8	14	22	40	68	115	181															
31500 - 49999	0	1	4	8	14	24	38	67	111	186																
50000 o más	2	3	7	14	25	40	63	110	181	301																

* Indica que el número de unidades de muestra procedente de los últimos 10 lotes o conjuntos no es suficiente para la inspección abreviada correspondiente a este AQL. En este caso, pueden usarse más de 10 lotes o conjuntos para el cálculo, siempre y cuando los lotes o conjuntos usados sean los más recientes en secuencia, que todos ellos hayan pasado por una inspección normal y que ninguno haya sido rechazado durante la inspección original.