

UNIVERSIDAD AUSTRAL DE CHILE
CAMPUS PUERTO MONTT
ESCUELA DE INGENIERIA EN COMPUTACION



ADMINISTRACIÓN DE APLICACIONES WIN32 A TRAVÉS DE SUS PROCESOS

Seminario de Titulación para optar
al título de Ingeniero en Computación

PROFESOR PATROCINANTE
Sra. VIVIANA ALVARADO ESPINOZA

PROFESOR COPATROCINANTE
Srta. CLAUDIA ZIL BONTES

Aris David Iturra Soto

PUERTO MONTT - CHILE
2006

a la memoria...

...a el espíritu Iturriano

...a la vida revolucionaria

...de mis grandes amigos

SINTESIS

En este seminario se plantean resultados cuantitativos del procedimiento que permite la administración de aplicaciones Win32 a través de sus procesos, y además se muestra otra forma de operar con éste, es decir, en vez de atacar procesos inválidos (programas maliciosos) eliminándolos del sistema luego de su instalación, se evita siquiera instalarlos.

A través del análisis de la situación actual en donde el computador se ve afectado constantemente por procesos inválidos y el planteamiento de una situación hipotética que incluya un procedimiento de control de procesos, se pretende, a través del contraste de ambas, determinar si la realidad hipotética es mejor que la realidad actual. Ahora bien, este procedimiento está determinado por una aplicación prototipo que trabaja de la forma hipotética planteada debido a que mantiene una lista de procesos válidos.

Con esta información: situación sin prototipo y situación con prototipo, es factible obtener la base para los diseños tanto lógico como físico de la aplicación prototipo. En el primero, diseño lógico, se incluirán los diagramas de flujos de datos y de estructura de los procesos involucrados y, el modelo Entidad Relación que se utilizará. En el segundo (diseño físico), se traducirá todo lo anterior de modo de crear la aplicación funcional.

Los pasos siguientes de la metodología incluyen el análisis del estado operacional de los PCs y su evaluación por medio de gráficos, para luego

comparar las estadísticas recopiladas antes de instalar el prototipo con las posteriores.

Cuantitativamente, el resultado de la comparación es favorable a la segunda realidad, por lo tanto, la aplicación funcional permite un mejor desempeño del computador, lo cual se ve reflejado en la reducción de los costos asociados a los tiempos de servicio técnico.

ABSTRACT

This work evaluates the results of using a method for administering Win32 applications through their processes. It also shows another approach to these methods: attacking non-valid processes (malicious programs) by not allowing them to be installed, rather than by removing them from the system after they have been installed.

By comparing the current situation, where computers are constantly affected by non-valid processes; with a new hypothetical situation, where we have a procedure for controlling these processes, we can determine which works best. This comparison is accomplished through a prototype application that works like the above-mentioned hypothetical by managing a list of valid processes.

The information that will be obtained from the comparison between the current situation and the situation we intend to create by using this prototype will provide the basis for logical and physical design for the prototype application. The logical design will include the data flow charts and the structure diagrams for the processes, as well as the entity-relationship model that will be used. The physical design will translate that into a functional application.

The following steps include analyzing the operational conditions of the PCs using graphs and comparing the statistics obtained before and after installing the prototype.

The results obtained from the comparison are more favorable with the prototype than with the present situation. The functional application allows the computer to function more efficiently, and reduces maintenance costs.

ÍNDICE

1.	INTRODUCCIÓN	1
2.	OBJETIVOS.....	3
2.1	OBJETIVO GENERAL:	3
2.2	OBJETIVOS ESPECÍFICOS:	3
2.3	PREGUNTAS DE INVESTIGACIÓN:.....	4
2.4	HIPÓTESIS:	4
3.	PLANTEAMIENTO DEL PROBLEMA	5
3.1	ANTECEDENTES	5
3.1.1	<i>Definición del Problema:.....</i>	5
3.1.2	<i>Identificación de Esfuerzos Anteriores para Resolverlo:.....</i>	5
3.1.3	<i>Solución Propuesta:</i>	8
3.1.4	<i>Equipo de Trabajo:.....</i>	9
3.2	JUSTIFICACIÓN:.....	10
3.2.1	<i>Situación sin Proyecto:.....</i>	10
3.2.2	<i>Situación con Proyecto:.....</i>	11
3.3	DELIMITACIÓN DEL PROBLEMA:.....	12
3.3.1	<i>Limitaciones Generales</i>	12
3.3.2	<i>Supuestos y Dependencias.....</i>	12
4.	METODOLOGÍA.....	13
4.1	MARCO TEÓRICO	13
4.2	EL MÉTODO	15
4.3	ESTRUCTURA METODOLÓGICA DEL PROYECTO	16
5.	RECURSOS.....	18
5.1	HARDWARE.....	18
5.2	SOFTWARE	19
6.	DIAGNOSTICO DE SITUACIÓN DE LOS PCS. Y ANÁLISIS Y DISEÑO DEL PROTOTIPO FUNCIONAL	21
6.1	DOCUMENTACIÓN DEL ENTORNO Y REQUISITOS DEL PROTOTIPO FUNCIONAL	22
6.1.1	<i>Estudio Inicial del Entorno Operacional de los PCs.....</i>	22
6.1.2	<i>Análisis del Estado Operacional I de los PCs.....</i>	26
6.1.3	<i>Construcción del Prototipo Funcional que Permita la Aplicación del Procedimiento.....</i>	30
6.2	ANÁLISIS.....	30
6.2.1	<i>Resumen de Requerimientos</i>	30
6.2.2	<i>Requerimientos Funcionales</i>	32
6.2.3	<i>Requerimientos Operacionales</i>	33
6.3	DISEÑO.....	35
6.3.1	<i>Estrategia de Flujos de Datos.....</i>	35
6.3.2	<i>Modelo Entidad Relación.....</i>	62
6.3.3	<i>Diseño Estructurado.....</i>	83
6.3.4	<i>Modelo Físico y Detalle de la Base de Datos</i>	91

7. IMPLEMENTACIÓN PERFECCIONAMIENTO DEL PROTOTIPO FUNCIONAL Y POSTERIOR ANÁLISIS DE LA SITUACIÓN DEL ENTORNO OPERACIONAL DE LOS PCS, UNA VEZ INSTALADO EL PROTOTIPO FUNCIONAL	95
7.1 DESARROLLO.....	95
7.1.1 <i>Codificación y Diseño de Interfaz</i>	95
7.1.2 <i>Implementación y Carga de Base de Datos</i>	107
7.2 PRUEBAS.....	115
7.2.1 <i>Pruebas del Software</i>	115
7.3 ANÁLISIS DEL ESTADO OPERACIONAL II DE LOS PCS.....	154
8. EVALUACIÓN Y COMPARACIÓN DE LAS PRUEBAS OBTENIDAS DE LOS ESTADOS OPERACIONALES (I Y II) DE LOS PCS.	156
8.1 EVALUACIÓN DEL ESTADO OPERACIONAL DE LOS PCS	156
8.2 COMPARACIÓN DE LAS PRUEBAS OBTENIDAS DE LOS ESTADOS OPERACIONALES (I Y II) DE LOS PCS.....	160
9. CONCLUSIONES Y/O RECOMENDACIONES	162
10. BIBLIOGRAFÍA	164
11. ANEXOS	165
11.1 ¿QUÉ ES UN PROCESO?	165
11.1.1 <i>Procesos y Subprocesos</i>	168
11.1.2 <i>Acerca de los Procesos y los Subprocesos</i>	168
11.2 MÉTODOS “INVESTIGACIÓN Y DESARROLLO” DE LA INFORMÁTICA.....	169
11.2.1 <i>Las Diversas Clases de Métodos de Investigación</i>	169
11.2.2 <i>Métodos Cuantitativos</i>	170
11.3 COMPONENTES Y SÍMBOLOS UTILIZADOS EN UN DIAGRAMA DE FLUJOS DE DATOS.	172
11.3.1 <i>Componentes en un DFD</i>	172
11.3.2 <i>Símbolos en un DFD</i>	173
11.3.3 <i>Nomenclatura en un diccionario de datos</i>	174
11.4 DESCRIPCIÓN DE FLUJOS DE DATOS DEL DIAGRAMA DE CONTEXTO....	175
11.5 DESCRIPCIÓN DE SIMBOLOGÍA UTILIZADA EN DIAGRAMA ENTIDAD RELACIÓN	177

TABLAS

Tabla N°1 Equipo de Trabajo	9
Tabla N°2 Tiempo que toma la reparación, medido en hora laborables, previa a la instalación del prototipo	22
Tabla N°3 Tiempo existente entre las mantenciones, medido en días laborables, previo a la instalación del prototipo	23
Tabla N°4 Costos por concepto de mantenciones, Sin Aplicación	24
Tabla N°5 Equipos infectados o que demuestren fallas por este motivo	24
Tabla N°6 Compra de nuevo hardware	24
Tabla N°7 Software no relacionado con el ámbito productivo, instalado por el usuario	25
Tabla N°8 Sumatoria del tiempo, en días laborales, que tarda la reparación de los equipos, vista por meses	26
Tabla N°9 Sumatoria de los tiempos, en días laborales, que tarda la mantención, por PCs	27
Tabla N°10 Promedio de días existentes entre mantenciones, vista por PCs	28
Tabla N°11 Sumatoria Costos por mantención según PC	29
Tabla N°12 Sumatoria Costos por mantención según Mes	29
Tabla N°13 Tiempo que toma la reparación, medido en hora laborables, prototipo ya instalado	154
Tabla N°14 Tiempo existente entre las mantenciones, medido en días laborables, prototipo ya instalado.....	155
Tabla N°15 Costos por concepto de mantenciones, Con Aplicación.....	155
Tabla N°16 Sumatoria de los tiempos, en días laborales, que tarda la mantención	156
Tabla N°17 Sumatoria del tiempo, en días laborales, que tarda la reparación de los equipos, vista por meses	157
Tabla N°18 Promedio de días existentes entre mantenciones, vista por PCs.	158
Tabla N°19 Sumatoria Costos por mantención según PC	159
Tabla N°20 Sumatoria Costos por mantención según Mes	159
Tabla N°21 Sumatoria Total del Tiempo en días laborales, que toman las Reparaciones	160
Tabla N°22 Tiempo existente entre las mantenciones, Promedio semestral, medido en días laborales	161
Tabla N°23 Sumatoria Total de los costos de Mantención	161

FIGURAS

Figura N°1 Metodología Empleada, Etapas y Sub-Etapas	17
Figura N°2 Diagrama de Contexto para el Sistema de Control de Procesos.	36
Figura N°3 Diagrama de nivel Superior para el Subsistema Equipo Computacional.....	38
Figura N°4 Diagrama de Nivel Superior para el Subsistema Control de Usuario.	38
Figura N°5 Diagrama de Nivel Superior para el Subsistema Tipos de Usuario.	38
Figura N°6 Diagrama de Nivel Superior para el Subsistema de Control de Procesos de Sistema Operativo.....	39
Figura N°7 Nivel 4, Subsistema de Control de Procesos de Sistema Operativo.	39
Figura N°8 Nivel 4.1, Ingresar Datos Proceso.	39
Figura N°9 Nivel 4.1.1, Validar Existencia de Proceso.	40
Figura N°10 Nivel 4.2, Asociar Procesos Validos.	40
Figura N°11 Nivel 4.2.1, Validar Existencia Asociación.	40
Figura N°12 Diagrama Entidad Relación Tentativo.....	66
Figura N°13 Modelo Conceptual Según PowerDesigner Trial 9.....	66
Figura N°14 Relación “Los Computadores Ejecutan Procesos de Sistema Operativo”.....	67
Figura N°15: Refinación “Los Computadores Ejecutan Procesos de Sistema Operativo”.....	67
Figura N°16 Diagrama Entidad Relación Final.....	79
Figura N°17 Validación de las transacciones en el diagrama Entidad Relación.....	80
Figura N°18 Diagrama de estructura Subsistema Control de Procesos de S.O.....	84
Figura N°19 Modelo Físico del sistema. según PowerDesigner Trial 9.....	91
Figura N°20 Tipo de datos de la tabla Tproces.....	92
Figura N°21 Tipo de datos de la tabla Computador.....	93
Figura N°22 Tipo de datos de la tabla Asociados.	93
Figura N°23 Tipo de datos de la tabla Perfil.....	93
Figura N°24 Tipo de datos de la tabla Características.....	94
Figura N°25 Tipo de datos de la tabla Tipo.....	94
Figura N°26 Diseño de Interfaz de Inicio de Sesión.....	104
Figura N°27 Diseño de Interfaz de Equipos.	104
Figura N°28 Diseño de Interfaz de Procesos y Aplicaciones y Equipos.....	105
Figura N°29 Diseño de Interfaz de Inserción y Eliminación de Usuarios....	105
Figura N°30 Diseño de Interfaz de Inserción y Eliminación de Usuarios....	106
Figura N°31 Diagrama de Caminos Independientes.....	116
Figura N°32 Diagrama de Caminos Independientes.....	133
Figura N°33 Diagrama de Caminos Independientes.....	144

Figura. N°34 Diferencia entre Programa y Proceso, http://www2.udec.cl/~elozada/capitulo II.htm	166
Figura. N°35 Vida de un Proceso, diseño para proyecto	166
Figura. N°36 Métodos de Investigación y Desarrollo	171
Figura. N°37 Símbolos utilizados en diagramas de flujos de datos.	173
Figura. N°38: Simbología utilizada en diagrama E-R.....	177

GRAFICOS

Gráfico N°1 Sumatoria del Tiempo, en días Laborales, que Tarda la Reparación de los Equipos, Vista por Meses	26
Gráfico N°2 Sumatoria de los Tiempos, en Días Laborales, que Tarda la Mantenición, por PCs.....	27
Gráfico N°3 Promedio de Días Existentes entre Mantenciones, Vista por PCs	28
Gráfico N°4 Sumatoria Costos por Mantención Según PC.....	29
Gráfico N°5 Sumatoria Costos por Mantención Según Mes.....	29
Gráfico N°6 Sumatoria de los Tiempos en Días Laborales, Periodo que Tarda la Mantención.....	156
Gráfico N°7 Sumatoria del Tiempo en Días Laborales que Tarda la Reparación de los Equipos, vista por Meses.....	157
Gráfico N°8 Promedio de Días Existentes entre Mantenciones, Vista por PCs	158
Gráfico N°9 Sumatoria Costos por Mantención según PC	159
Gráfico N°10 Sumatoria Costos por Mantención según Mes.....	159
Gráfico N°11 Sumatoria Total del Tiempo en Días Laborales, que Toman las Reparaciones	160
Gráfico N°12 Tiempo Existente entre las Mantenciones, Promedio Semestral, Medido en Días Laborales	161
Gráfico N°13 Sumatoria Total de los costos de Mantención.....	161

1. INTRODUCCIÓN

Quienes trabajan con computadores personales, en lugares como oficinas o laboratorios computacionales, más de alguna vez se han topado con problemas de mal funcionamiento del software, debido a fallas asociadas a la sobrecarga de aplicaciones en los PCs.

Si se excluye la posibilidad de falla del software, por instalación o errores internos de programación, se puede encontrar la respuesta a la pregunta anterior; en su uso e instalación. Debido a que es el usuario quien necesita de programas para realizar su gestión diaria y muchas veces éste no solicita ni cuenta con el apoyo de un perito, se puede deducir, que a causa de la falta de conocimiento del alcance de las acciones, es éste quien corrompe el sistema, logrando con ello la falla en el software. Entonces, ¿qué acciones, voluntarias o involuntarias, realizadas por el usuario, provocan fallas en las aplicaciones?

La fuerza productiva, comunicativa y estructural que reside en la versatilidad de capacidades que componen un computador, es lo que lleva al usuario a realizar actividades tales como: instalar aplicaciones, descargar música, jugar, chatear, etc. Sumado a lo anterior, los usuarios se ven afectados por programas maliciosos virus y malware que con o sin previo aviso, se instalan en las computadoras, lo cual, a corto o largo plazo llevará también a que el software no sea soportado por el hardware.

Es aquí donde cabe detenerse y reflexionar: ¿qué hacer para minimizar la ocurrencia de errores asociados a la instalación por omisión, falta conocimiento, o implantación de software dañino?

Esta es la interrogante que se pretenderá resolver a través del desarrollo del presente trabajo de tesis.

2. OBJETIVOS

2.1 Objetivo General:

Desarrollar un procedimiento, traducido en una aplicación, para la Administración Win32 a través de sus procesos, que permitirá reducir costos y tiempo de mantención de equipos informáticos. El enfoque de desarrollo se orienta a cambiar la forma de operar tradicional del procedimiento Administración de Aplicaciones Win32.

2.2 Objetivos Específicos:

- Reforzar algunas tareas de administración y control del ingeniero en computación, mediante la aplicación del procedimiento “Administración de aplicaciones win32 a través de sus procesos”.
- Establecer el propósito que cumplen operacional o productivamente, todos aquellos computadores, con los se trabaje en la tesis.
- Desarrollar un prototipo funcional, cliente-servidor, que permita demostrar la viabilidad del objetivo general.
- Evaluar el impacto que causa la aplicación del objetivo general.
- Administrar las aplicaciones remotamente por medio del prototipo funcional.

- Reducir los software del tipo dañino (virus, adwares, troyanos, etc) y de aquellos que no cumplan los criterios (seguridad, productividad etc.) definidos por el o los usuarios.

2.3 Preguntas de Investigación:

¿Cuál es el impacto que causa el cambiar el paradigma de operación del procedimiento de Administración de aplicaciones win32 a través de sus procesos, sobre la gestión del ingeniero, y cómo esto altera las operaciones productivas de los usuarios?

2.4 Hipótesis:

HP1: La aplicación del objetivo general mediante un prototipo funcional, permitiría disminuir los costos asociados a las mantenciones de los PCs, al definir medios de control y administración sobre éstos.

HP2: La aplicación del objetivo general, permitiría ampliar los tiempos entre mantenciones.

3. PLANTEAMIENTO DEL PROBLEMA

3.1 Antecedentes

3.1.1 *Definición del Problema:*

La constante escasez de recursos y la necesidad de optimizar los existentes, tanto a nivel de empresas como de individuos, hacen necesaria una acción procedimental, frente a problemas asociados al manejo de las aplicaciones que instala el usuario, o las que se instalan producto de infiltraciones por los bugs (hoyos de seguridad) del sistema operativo, virus o malwares, que obligan a la realización de constantes mantenciones del software, actualizaciones de hardware o la compra de insumos con el fin de mejorar el desempeño de éstos.

3.1.2 *Identificación de Esfuerzos Anteriores para Resolverlo:*

Un breve análisis de la perspectiva del uso del procedimiento de “Administración de aplicaciones win32 a través de sus procesos”, permite apreciar el modo de empleo, el cual se orienta siguiendo el enfoque de: “impedir la ejecución de todo proceso de aplicación que se definan como bloqueadas, permitiéndose el normal funcionamiento del resto de los

softwares. La problemática se plantea desde el punto de vista de discriminar aquellas aplicaciones que se deseen bloquear.

Buscando resolver los problemas planteados en el punto anterior, la computación como ciencia, ha creado un conjunto de reglas que definen sus límites, estableciendo lo debido para tener éxito dentro de estos, permitiéndole explicar y predecir su comportamiento. Bajo esta premisa, la operatoria aplicada al marco de soluciones, va de la mano con la aparición de los problemas. Eso quiere decir, que debe aparecer un problema, para plantearse la búsqueda de la solución. Ejemplo de ello, los antivirus informáticos, una vez que aparece un virus informático se crea un antivirus.

Los esfuerzos en este sentido han sido dirigidos a:

- Prevenir y controlar las acciones puntuales que afectan al software.
- Mejora de Hardware

Otra forma tradicional de resolver los problemas planteados en el punto 3.1.1 está dada por la limitación de los privilegios del usuario, (prestaciones que ofrece el sistema operativo), frente al uso del software, lo cual define que puede o no hacer éste. La problemática es que muchos programas se saltan estas limitaciones permitiendo con ello que se ejecuten o modifiquen partes sensibles del sistema operativo.

Otra forma es instalar un software que limite las acciones de los demás softwares (antivirus, firewalls, antispywares). Este, normalmente, detecta las

modificaciones en el sistema operativo, de forma que cuando ello ocurra, informe al usuario y le sugiera las medidas a seguir (bloquear, permitir, eliminar o enviar a cuarentena). Esta metodología es, a lo menos, discutible debido a que entre tantas opciones el usuario inexperto se confunde y finalmente oprime cualquier botón que le permite salvar la situación.

Como medida complementaria a lo mencionado anteriormente, se opta por comprar o mejorar el hardware existente, a fin de optimizar las prestaciones de los programas, buscando solucionar los problemas planteados en el punto 3.1.1.

3.1.3 Solución Propuesta:

Implementar una solución que busque optimizar los recursos informáticos y los impactos económicos que estos generan, no pasa por identificar y eliminar los problemas asociados a la ralentización que sufren los computadores por la sobrecarga de aplicaciones, derivados de la instalación uso y ejecución de estos (se parte de premisa que los software están bien construidos), si no más bien en limitar sus fuentes para minimizar su ocurrencia. Esto hace necesario definir el marco de operación, limitándolo a una gestión adecuada y eficiente de las aplicaciones.

Para realizar lo anterior, se plantea como solución la elaboración de un procedimiento, que administre las aplicaciones win32 a través de sus procesos.

Esta propuesta contempla el desarrollo de un prototipo funcional, entendiéndose como prototipo funcional a una aplicación terminada, con todas las funcionalidades que permiten alcanzar los objetivos planteados, pero que posteriormente pueden agregarse más funcionalidades. Este prototipo se encargaría de bloquear todas las aplicaciones, aceptando sólo la ejecución de aquellas que se encuentren en un listado de software permitidos.

Esto se realizaría monitoreando los procesos que se ejecuten en un determinado computador, con el fin de contrastarlos con la lista preexistente de aplicaciones permitidas.

La decisión sobre que programas constituirán esta lista, será responsabilidad del administrador del sistema, pudiendo ser este el propio usuario o el encargado informático de una empresa, según los requerimientos y la actividad realizada sobre el computador.

3.1.4 Equipo de Trabajo:

Tabla N°1 Equipo de Trabajo

Desarrollador	Aplicación
Aris Iturra	Administración de aplicaciones a través de sus procesos.

3.2 Justificación:

3.2.1 Situación sin Proyecto:

Los usuarios comunes de hoy se ven afectados por la mantención regular de sus equipos, lo cual conlleva tiempos de inoperancia y provoca una disminución de la producción, en el recurso humano. Por otro lado, este hecho impacta en los recursos económicos, toda vez que se debe incurrir en un costo por mantención técnica.

El avance en las comunicaciones e Internet, ha permitido que se masifiquen los programas maliciosos y virus, los cuales actúan sin previo aviso y apoyados en la ingenuidad del usuario. Este último, muchas veces acepta la instalación de aplicaciones sin leer los mensajes de advertencia.

Lo anterior influye en la ralentización que sufren los equipos. Cuando el usuario percibe que la ralentización es muy alta, lo más frecuente es que se opere de las siguientes tres formas:

- Envío del equipo a mantención, aquí usualmente se reinstala el sistema operativo y aquellas aplicaciones consideradas por el usuario.
- Instalación y actualización de un antivirus, aumentándose muchas veces la ralentización.
- Mejora del hardware del PC o compra de un nuevo PC.

3.2.2 Situación con Proyecto:

La solución propuesta, buscaría limitar la ocurrencia de los problemas, mencionados en el punto 3.1.1; relacionando las actividades que se pueden ejecutar en un determinado PC, al seleccionar las aplicaciones y asociarlas con las actividades que debiesen realizarse en éste.

La idea es habilitar sólo aquellas aplicaciones autorizadas negándose todas las demás, mediante el control de los procesos que genera la ejecución de cualquier aplicación. Lo anterior permitiría minimizar las mantenciones, en el sistema operativo, producto de instalaciones o ejecución de software inapropiado, la implantación de softwares dañinos (virus, malware, spywares, gusanos, entre otros), la disminución de los tiempos ociosos que trascurren mientras el o los computadores se encuentran en proceso de mantención.

Esto debería potenciar y facilitar la labor del administrador, permitiendo una mejor gestión al hacerla más eficiente, producto del manejo y registro que se tendría de los software en cada equipo, además aumentarían las prestaciones del hardware en el tiempo, dado que se minimizaría el problema de la ralentización, lo que debiera conducir a la disminución de los gastos asociados con los ítems de actualización del PC.

3.3 Delimitación del Problema:

3.3.1 *Limitaciones Generales*

- La solución partirá de cero por lo que no estará sujeta a otros estándares o regulaciones de antiguos sistemas.
- Se trabajará gestionando sólo los procesos primarios (ver concepto en anexo 11.1.3).
- Se han de cumplir los requerimientos mínimos de equipamiento y software de sistema para una correcta ejecución de la solución.

3.3.2 *Supuestos y Dependencias*

Se han establecido los siguientes supuestos:

- Los equipos en los que se va a ejecutar la solución, deben cumplir los requisitos antes indicados, para garantizar una ejecución correcta de la misma.
- El usuario, sólo necesita conocer y manejar los distintos parámetros de configuración del software con los que trabaje.

4. METODOLOGÍA

4.1 Marco Teórico

Según la [RAE] metodología es el “*Conjunto de métodos que se siguen en una investigación científica o en una exposición doctrinal*”, de lo anterior se puede extraer que el objetivo de cualquier metodología es estructurar la forma de adquirir o exponer los conocimientos. Para llevar a cabo esta misión se debe establecer el significado de método. Un método es un “*Procedimiento que se sigue en las ciencias para hallar la verdad y enseñarla*” [RAE];. por tanto es fundamental la elección del método adecuado para permitir conocer la realidad.

La [RAE] divide las ciencias¹ en muchas y muy variadas clasificaciones. De acuerdo a esta, no se puede encuadrar claramente a las ingenierías dentro de una de estas clasificaciones, no porque no sea una aplicación directa de una única *ciencia pura*², sino porque las ingenierías, no son mera aplicación de otras ciencias.

¹ El diccionario de la RAE (200) divide las ciencias en: *exactas* (matemáticas), *humanas o sociales* (psicología, sociología, filosofía, etc.) y *naturales* (botánica, zoología, etc. y a veces, también se incluyen aquí la física y la química). El RAE habla también de ciencias *ocultas* (magia, alquimia, astrología, etc.).

² Bunge, M. *La Investigación Científica*. Ed. Ariel, S.A. Barcelona, 1976.

[Aracil]³ en su ensayo de “Máquinas, sistemas y modelos” es quien fija el marco de referencia que permite interrelacionar la ciencia con la ingeniería, este ensayo postula: *“las ciencias se ocupan de lo natural, mientras que el dominio específico de la ingeniería es lo artificial”*. Sin embargo, no es él único que reclama la necesidad de definir una “ciencia de la ingeniería”, así por ejemplo [Blum]⁴, detalla, en *“Beyond Programming”*, que la ciencia de la tecnología de la computación es *“...el estudio de la transformación de ideas en operaciones”*.

Si se entiende que un procedimiento es una operación en sí, el enfoque de [Blum] permite establecer la naturaleza de esta investigación, contribuyendo a adoptar la metodología de análisis de este estudio.

³ Aracil, J. Máquinas, sistemas y modelos. Un ensayo sobre sistemática. TECNOS, S.A. Madrid, 1986.

⁴ Blum, B. I. *Beyond Programming: To a New Era of Design*. Oxford University Press, 1996.

4.2 El Método

El objetivo de cualquier ciencia es adquirir conocimientos y la elección del método adecuado que permita conocer la realidad es por tanto fundamental.

Los métodos son los procedimientos que permiten captar los fenómenos o resolver los problemas relacionados con ellos⁵. Según se trate de procesos de investigación, desarrollo, aplicación, evaluación de los objetos de la Informática (sistemas, algoritmos, etc.) se toma con mayor énfasis los rasgos metodológicos, que en general responden al proceder del trabajo científico ya que sus conocimientos responden a las exigencias de la universalidad, objetividad, reproducción, etc.

De esta forma, los Sistemas de Información, se vinculan con las ciencias empíricas al tratar de explicar la realidad estableciendo referentes con la Informática, al utilizar el método “CUANTITATIVO”, y que al hacer frente a los procedimientos de tipo técnico la vinculan a la realidad concreta.

La metodología de investigación aplicada a este proyecto, se basa en el método experimental controlado de análisis dinámico, que a través de un producto desarrollado mide la performance⁶

⁵ Graciela E. Barchini, Mabel Sosa, Susana Herrera. La informática como disciplina científica. Ensayo de mapeo disciplinar

⁶ Graciela E. Barchini. Métodos “Investigación y desarrollo” de la Informática

4.3 Estructura Metodológica del Proyecto

El ingeniero es un ente que se ve enfrentado a diversas situaciones. De ahí, que la estrategia metodológica implementada se base en procesos inventivos e innovadores.

Desde el momento que se selecciona el tema, hasta el momento que se redacta el informe final, la investigación pasa por distintos períodos a lo largo de su desarrollo. Estos diferentes períodos serán llamados etapas.

Una etapa es un segmento de ejecución de trabajo interdependiente, que por sus características se distingue de otros. Las etapas se relacionan entre sí constantemente, a través de las observaciones, de modo de proporcionar información que luego es incorporada a la investigación en forma de evidencias de su veracidad.

El plantearse el análisis del impacto que causa el cambiar la forma de operación del procedimiento, “Administración de aplicaciones win32 a través de sus procesos”, sobre la gestión del ingeniero y cómo esto altera las operaciones productivas de los usuarios, conlleva definir las etapas y los mecanismos de control que permiten evaluar el comportamiento que adquiere el procedimiento de acuerdo a su uso. Este planteamiento, obliga a establecer un diseño metodológico desagregado por etapas conforme a una serie de pasos, que permitan identificar claramente los factores bajo estudio

y analizar en forma ordenada y sistemática sus componentes del modo más racional. Ver figura N°1.

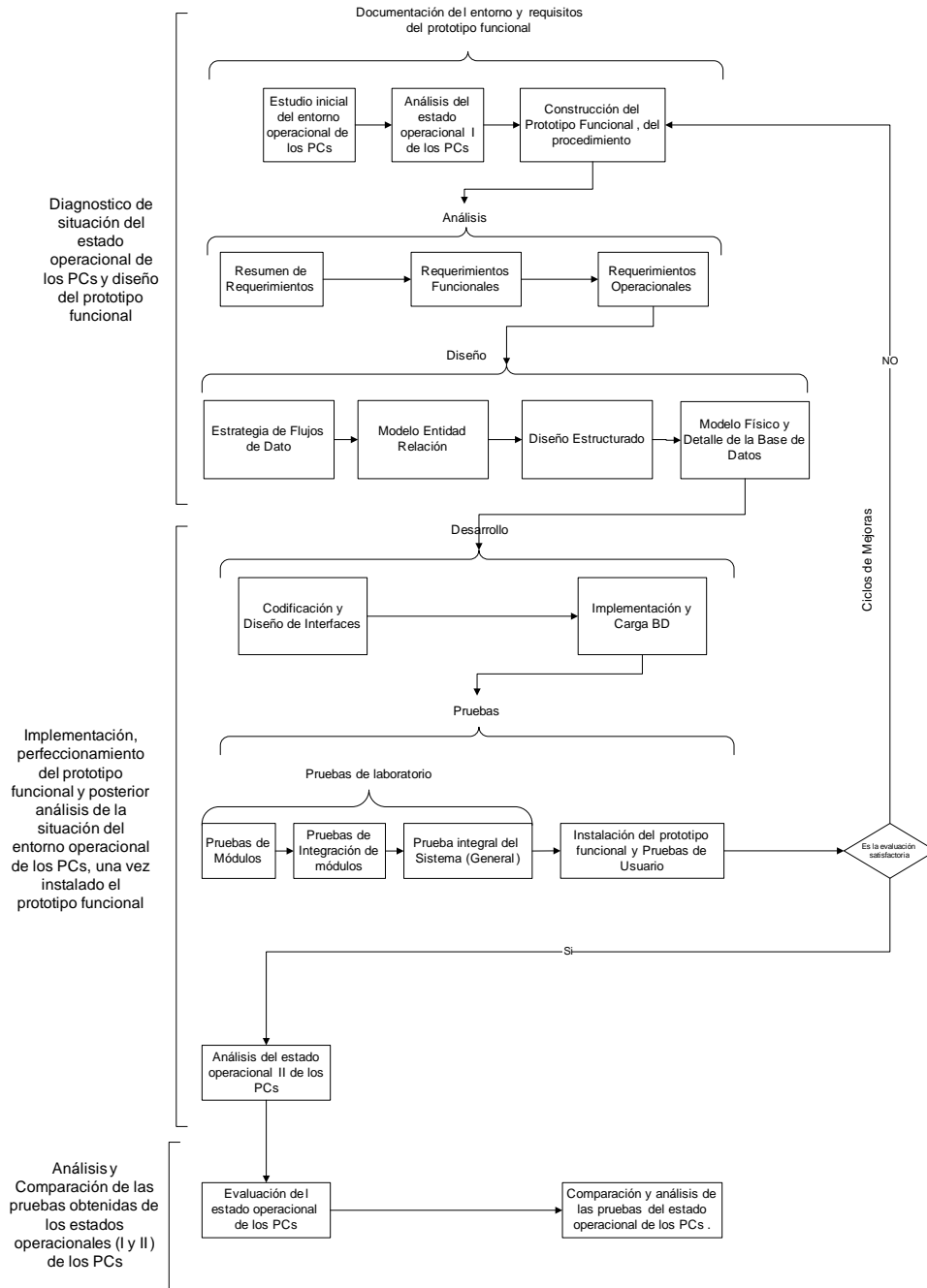


Figura N°1 Metodología Empleada, Etapas y Sub-Etapas

5. RECURSOS

5.1 Hardware

La elección del hardware de desarrollo está dado por uno de los objetivos intrínsecos del proyecto, optimizar los recursos existentes disminuyendo los gastos del cliente final. En resumen se busca evitar compra o actualización de nuevo hardware tanto en el desarrollo como en la implementación.

- Servidor

Pentium 4 : 1.8 Ghz

RAM : 256 MB

Disco Duro : 7200 RPM, 40 GB con a lo menos 100 MB libres.

Tarjeta de red.

- Clientes

Pentium III : 800 Mhz

RAM : 128 MB de Ram

Disco Duro : 7200 RPM, 10 GB con a lo menos 100 MB libres.

Tarjeta de red.

5.2 Software

Este proyecto de tesis busca desarrollar, implementar e implantar aplicaciones que requieran una menor demanda del hardware, minimizando el uso de los recursos del sistema operativo.

En cuanto a las tecnologías de información se debe trabajar con las herramientas con las que se cuenta con licencia. Debido a ello la base de datos a utilizar será Microsoft Access 97.

La elección de las herramientas de desarrollo Visual Basic 6.0 y Visual C++, esta dada por la flexibilidad e integración que estas tienen con los sistemas operativos win32 de Microsoft, su buena documentación, y la rapidez que otorga el hecho de ser lenguajes de tercera generación.

Los requisitos de software indispensables para el correcto funcionamiento del prototipo se definen a continuación:

Requerimientos de Software

Servidor y equipo de desarrollo

Lenguajes de programación:

Visual Basic 6.0 (Aplicaciones cliente y reconocimiento)

Visual C++ (para compilar las fuentes de una librería dinámica).

Motor de base de datos:

Microsoft Acces 97

Arquitectura y Plataformas:

Microsoft Windows con sistemas operativos de 32 bit, Windows 2000, Windows XP y Windows 2003.

- Clientes

Motor de base de datos:

Microsoft Acces 97

Arquitectura y Plataformas:

Microsoft Windows con sistemas operativos de 32 bit, Windows 2000, Windows XP y Windows 2003.

6. DIAGNOSTICO DE SITUACIÓN DE LOS PCs. Y ANÁLISIS Y DISEÑO DEL PROTOTIPO FUNCIONAL

El objetivo que se persigue en esta etapa es identificar los requerimientos, para ello en esta instancia, se hace necesario tener claro el trabajo a realizar y los recursos involucrados. Con el fin de cumplir esta meta, se deben identificar el o los criterios de análisis

El identificar el o los criterios de análisis conlleva diagnosticar la situación. Para ello, lo primero que se debe tener en claro son las variables representativas que se van a analizar a la hora de evaluar los factores que resultan del uso del procedimiento, “administración de aplicaciones win32 a través de sus procesos”.

Con el objetivo de establecer tales factores, se definen las siguientes variables representativas:

- Tiempo no operativo que toma la reparación previo a la instalación del prototipo
- Periodo existente entre mantenciones
- Costos por concepto de mantenciones
- Software no relacionado con el ámbito productivo
- Equipos infectados o que demuestren fallas por este motivo
- Compra de nuevo hardware

Para efectos del desarrollo de esta tesis los valores de los factores fueron obtenidos de lugares distintos, en los cuales se muestrearon ciertos PCs. que mostrasen relevancia para las empresas.

6.1 Documentación del Entorno y Requisitos del Prototipo Funcional

6.1.1 Estudio Inicial del Entorno Operacional de los PCs

Tabla N°2 Tiempo que toma la reparación, medido en horas laborables, previa a la instalación del prototipo

Equipo	mes1	mes2	mes3	mes4	mes5	mes6
Pc1	8	0	24	0	0	6
Pc2	0	0	12	0	3	0
Pc3	16	24	16	0	16	8
Pc4	0	8	8	0	6	0

Tabla N°3 Tiempo existente entre las mantenciones, medido en días laborables, previo a

la instalación del prototipo

PC1	Fechas	D Mantenciones
Mes1	24/01/2005	0
Mes3	17/03/2005	39
Mes6	09/06/2005	59
Promedio		33

PC2	Fechas	D Mantenciones
Mes3	14/03/2005	0
Mes5	13/05/2005	44
Promedio		22

PC3	Fechas	D Mantenciones
Mes1	07/01/2005	0
Mes2	28/02/2005	37
Mes3	03/03/2005	4
Mes5	11/05/2005	49
Mes6	20/06/2005	28
Promedio		24

PC4	Fechas	D Mantenciones
Mes2	24/02/2005	0
Mes3	17/03/2005	16
Mes5	25/05/2005	48
Promedio		21

Tabla N°4 Costos por concepto de mantenencias, Sin Aplicación

PC1	Fechas	Costo	Observación
Mes1	24/01/2005	20.000	Limpieza de Software en el Computador e Instalación de Antivirus
Mes3	17/03/2005	40.000	Instalación de S/O y Aplicaciones
Mes6	09/06/2005	15.000	Reinstalación Sistema Pesas
Total		75.000	

PC2	Fechas	Costo	Observación
Mes3	14/03/2005	20.000	Desinstalación de Software instalación de Antivirus
Mes5	13/05/2005	14.000	Modificación de Registros
Total		34.000	

PC3	Fechas	Costo	Observación
Mes1	07/01/2005	10.000	Limpieza Software y Actualización de antivirus
Mes2	28/02/2005	35.000	Instalación de Disco Duro Traspaso de Información
Mes3	03/03/2005	15.000	Actualización de Software del Sistema Contable
Mes5	11/05/2005	0	Garantía
Mes6	20/06/2005	12.000	Instalación de Memorias RAM
Total		72.000	

PC4	Fechas	Costo	Observación
Mes2	24/02/2005	7.000	Actualización de Antivirus
Mes3	17/03/2005	35.000	Instalación de Disco Duro y Memoria RAM Traspaso de Información
Mes5	25/05/2005	10.000	Mantención Software Pesas
Total		52.000	

Tabla N°5 Equipos infectados o que demuestren fallas por este motivo

Nombre del Software	PC1
FunLove	

Tabla N°6 Compra de nuevo hardware

PC3	Observación
Mes2	Instalación de Disco Duro Traspaso de Información
Mes6	Instalación de Memorias RAM

PC4	Observación
Mes5	Mantención Software Pesas

Tabla N°7 Software no relacionado con la labor que desempeñan los PCs y que es

instalado por los usuarios con fines recreativos

Nombre del Software	PC1
Eage of Empires I	
Ares	
Starcraf	
mesenger	
kaza lite	
edonky	
Sonic	
Winamp 3.0	
k-lite Codec Pack	
Macromedia Flash	
Bsplayer	

Nombre del Software	PC2
kaza lite	
Ares	
mesenger	
Comandos	

Nombre del Software	PC3
kazza	
mesenger	
yahoo mesenger	
Starcraf	
Comandos	
conejo lector	
Photoshop 7	

Nombre del Software	PC4
messenger	
yahoo mesenger	

6.1.2 Análisis del Estado Operacional I de los PCs

Tabla N°8 Sumatoria del tiempo, en días laborales, que tarda la reparación de los equipos, vista por meses

	Mes 1	Mes 2	Mes 3	Mes 4	Mes 5	Mes 6
PC-Sum-1	24	32	60	0	25	14

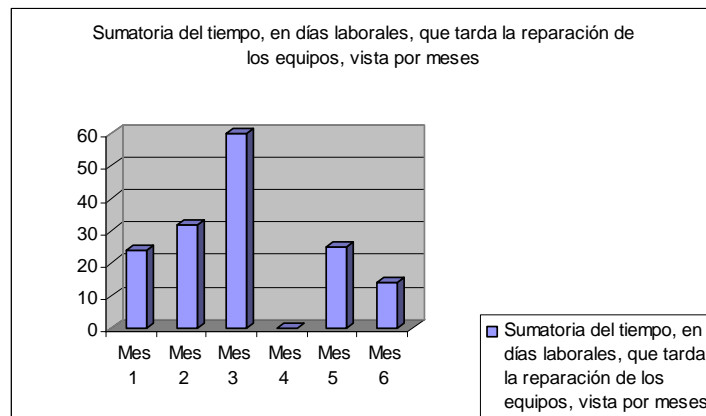


Gráfico N°1 Sumatoria del Tiempo, en días Laborales, que Tarda la Reparación de los Equipos, Vista por Meses

Tabla N°9 Sumatoria de los tiempos, en días laborales, que tarda la mantención, por PCs

Primer Semestre	
Pc1	38
Pc2	15
Pc3	80
Pc4	22

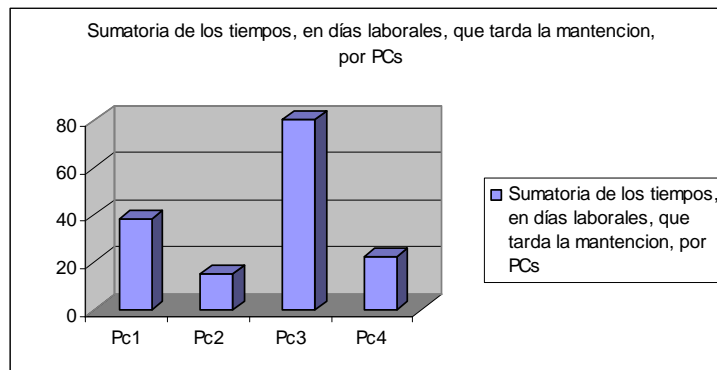


Gráfico N°2 Sumatoria de los Tiempos, en Días Laborales, que Tarda la Mantención, por PCs

Tabla N°10 Promedio de días existentes entre mantenciones, vista por PCs

Primer Semestre	
PC1	33
PC2	22
PC3	24
PC4	21

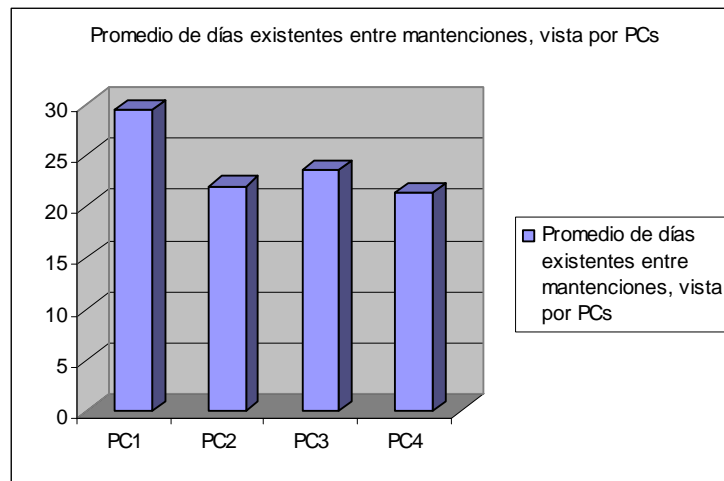


Gráfico N°3 Promedio de Días Existentes entre Mantenciones, Vista por PCs

Tabla N°11 Sumatoria Costos por mantención según PC

Primer Semestre	
PC1	75.000
PC2	35.000
PC3	72.000
PC4	52.000

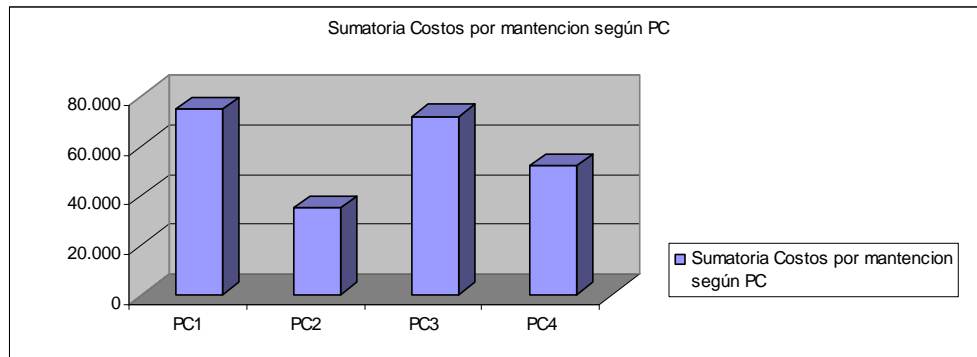


Gráfico N°4 Sumatoria Costos por Mantención Según PC

Tabla N°12 Sumatoria Costos por mantención según Mes

	Mes 1	Mes 2	Mes 3	Mes 4	Mes 5	Mes 6
Total por meses	30.000	42.000	110.000	0	24.000	27.000

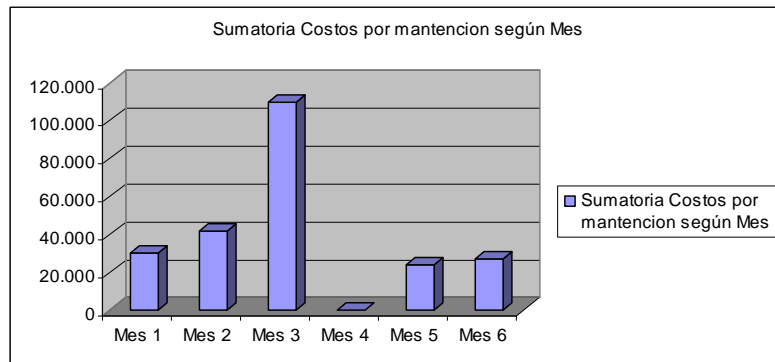


Gráfico N°5 Sumatoria Costos por Mantención Según Mes

6.1.3 Construcción del Prototipo Funcional que Permita la Aplicación del Procedimiento

6.2 Análisis

En este capítulo se explicarán los requerimientos necesarios para desarrollar el prototipo funcional, que permite llevar a cabo esta investigación.

6.2.1 Resumen de Requerimientos

Los requerimientos nacen a partir de las observaciones hechas en los distintos lugares donde se ha desarrollado actividades de mantención y soporte. Para determinar en forma clara y precisa los requerimientos que se deben considerar para que el prototipo cumpla con el propósito de la tesis, se han identificado los siguientes puntos:

El prototipo estará dividido en una aplicación cliente y servidor.

1. Deberá contar con una base de datos que contenga toda la información pertinente a procesos, aplicaciones, usuarios, y otros.
2. Otorgar un acceso restringido al administrador o administradores para obtener seguridad frente a irrupciones indeseadas.

3. El administrador realizará el mantenimiento del sistema agregando, modificando o borrando, los usuarios, los PCs, o los procesos y aplicaciones que éste controla.
4. Implementar menús y módulos a través de los cuales el administrador tendrá diversas opciones para acceder y manipular las opciones que ofrece el prototipo de aplicación.
5. Permitir al administrador cambiar su contraseña para otorgarle mayor seguridad a la hora de manipular datos.
6. La aplicación cliente podrá funcionar en forma autónoma (previa configuración) o administrada remotamente.
7. La aplicación contendrá una base de datos que estará replicada en todos los equipos clientes, actualizándose desde el servidor sólo cuando éste se encienda.
8. La aplicación cliente deberá permanecer oculta a los usuarios
9. La aplicación cliente debe monitorear constantemente la creación de procesos producto de la ejecución de aplicaciones por parte de los usuarios, sistema operativo u otras aplicaciones.

6.2.2 Requerimientos Funcionales

Dado el resumen de requerimientos anteriormente descrito, los requerimientos funcionales del sistema son los siguientes:

Al administrador deberá permitir:

- Autenticación del administrador.
- Listar menú de administración.
- Gestión, mantenimiento de la base de datos.
- Ingreso de procesos y asociación de éstos con su respectiva aplicación.
- Eliminación de procesos y asociación de éstos con su respectiva aplicación.
- Ingreso de nuevos usuarios con privilegios de administrador.
- Eliminar usuarios con privilegios de administrador.
- Seleccionar los equipos a monitorear.
- Asociar los equipos con las aplicaciones que éstos pueden ejecutar.

Los administradores podrán en los equipos clientes:

- Desplegar una interfaz oculta a través de una combinación de teclas
- Ingresar a través de esta interfaz mediante un nombre de usuario y una clave.

- Obtener una lista por pantalla de los procesos no permitidos que se intenten ejecutar en ese equipo.

6.2.3 *Requerimientos Operacionales*

Al momento de analizar los requerimientos de operación en el análisis de un sistema, se debe tener en cuenta lo que se espera o se necesita en términos de presentación y funcionalidad de éste. Por ello, los requerimientos esenciales para el desarrollo de la aplicación cliente servidor son:

- Conectividad

Es necesario que la aplicación cliente descargue las actualizaciones (aplicaciones que puede ejecutar) desde la base de datos que se encuentra en el servidor.

Para obtener este tipo de conectividad es necesario trabajar el sistema bajo la plataforma cliente servidor. Este último contendrá la base de datos de las aplicaciones permitidas, siendo accedido varias veces por todos aquellos PCs que cuenten con la aplicación cliente que se encuentren en la misma red del servidor.

- Interfaz amigable

La interfaz debe ser de fácil comprensión y con alta capacidad intuitiva, a fin de permitir una rápida asociación de las aplicaciones con sus procesos, dado que ésta puede ser ejecutada por usuarios en sus casas o en equipos personales, quienes querrán tener control sobre las actividades y sobre sus equipos. Por lo tanto se debe tener en cuenta, que no todos los usuarios que acceden a esta aplicación, son usuarios avanzados.

- Facilidad de operación y usabilidad

Los módulos deben permitir a los usuarios un desplazamiento fluido por el programa, para ello deben contar con una estructura bien elaborada para que el usuario no se pierda.

- Rapidez en el monitoreo de los procesos

Para una carga rápida del sistema operativo, el módulo que monitorea los procesos debe evitar el uso excesivo de procesador, así como las consultas a la base de datos. Los módulos deberán ser planeados a fin de contener el mínimo número de instrucciones y estar asociadas a instrucciones de más bajo nivel.

- Reutilizable para futuras versiones

Los sistemas deben ser capaces de ser reutilizados para posteriores versiones. Por tal motivo el estilo de programación ha de ser ordenada y con una notación de fácil comprensión para futuras y nuevas versiones del sistema.

6.3 Diseño

En esta etapa se desarrollará un modelado del sistema, a partir de herramientas como: diagrama de flujo de datos, modelo entidad relación y diagramas de estructuras.

6.3.1 Estrategia de Flujos de Datos

El Diagrama de Flujos de Datos (DFD) es la técnica que será la base en la construcción del modelo Entidad Relación y para definir los componentes que se utilizarán en la construcción del código.

La descripción de estos componentes y los símbolos utilizados en el diagrama de flujos de datos se muestran en el anexo 11.3.

6.3.1.1 Diagrama de Contexto

El diagrama de contexto es el camino más corriente de modelar el problema en su conjunto como un proceso único. En él se muestran todas las entradas que interactúan con el sistema y sus correspondientes flujos de información.

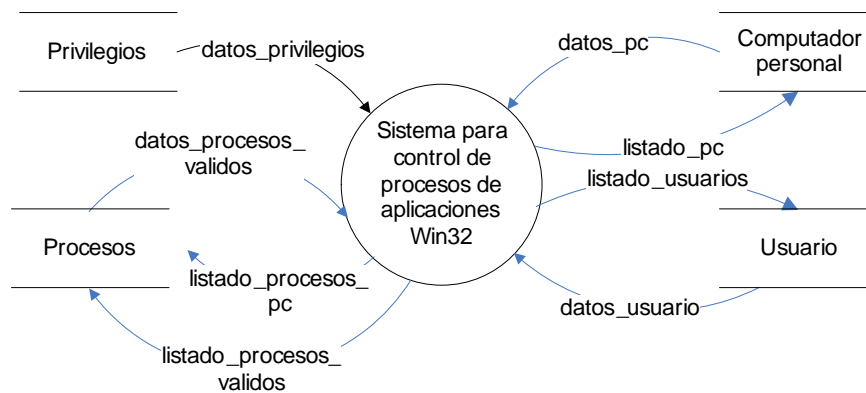


Figura N°2 Diagrama de Contexto para el Sistema de Control de Procesos.

Dado que un diagrama de contexto no describe en detalle el sistema, para obtener una mayor especificación, será necesario identificar los diagramas de nivel superior y luego expandirlos hasta llegar a un nivel que describa con mayor definición los flujos involucrados.

Los diagramas de nivel superior identificados son: Subsistema Equipo computacional, Subsistema Usuario, Subsistema Privilegios y Subsistema Procesos.

Como el Subsistema Procesos es el que reviste mayor complejidad será el único expandido al nivel que describa con mayor detalle los flujos de entrada y salida.

6.3.1.2 Diagramas de Nivel Superior.

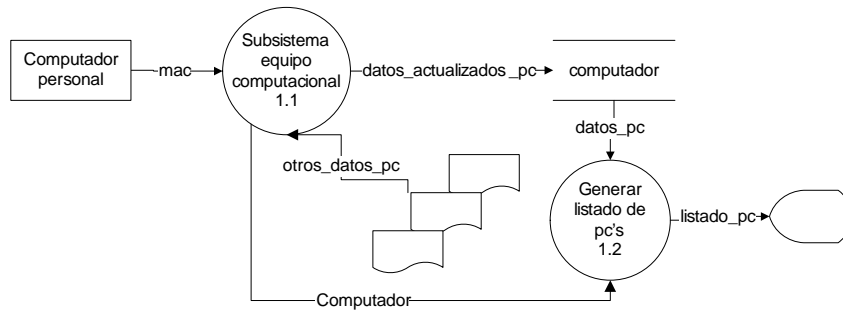


Figura N°3 Diagrama de nivel Superior para el Subsistema Equipo Computacional

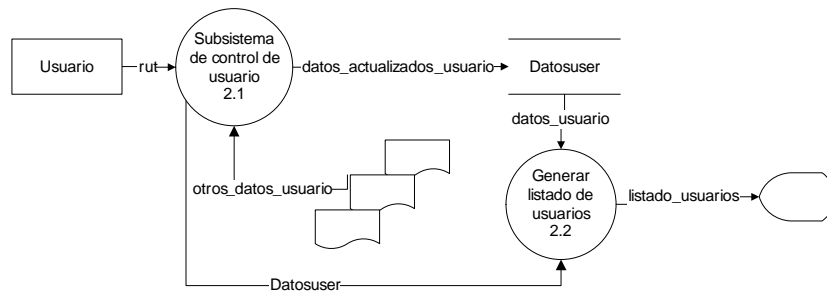


Figura N°4 Diagrama de Nivel Superior para el Subsistema Control de Usuario.

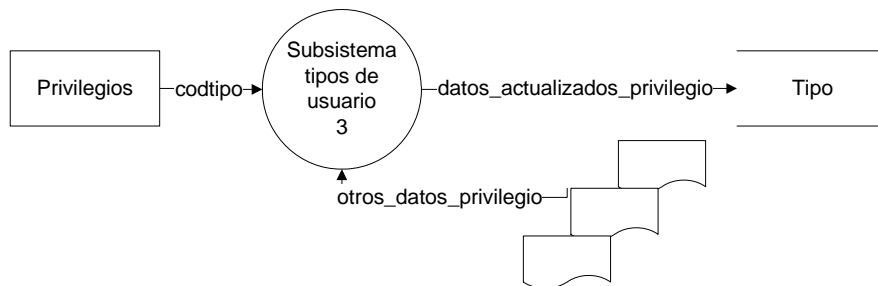


Figura N°5 Diagrama de Nivel Superior para el Subsistema Tipos de Usuario.

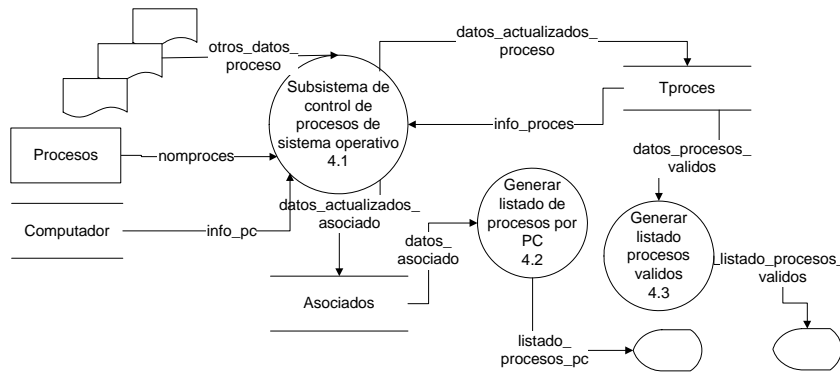


Figura N°6 Diagrama de Nivel Superior para el Subsistema de Control de Procesos de Sistema Operativo.

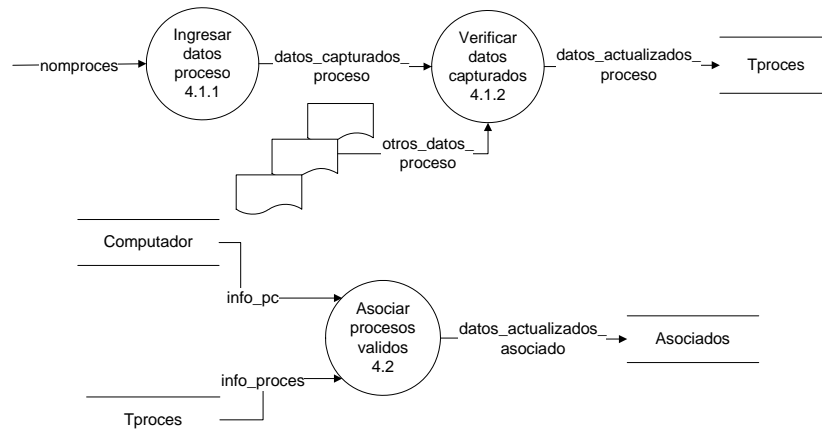


Figura N°7 Nivel 4, Subsistema de Control de Procesos de Sistema Operativo.

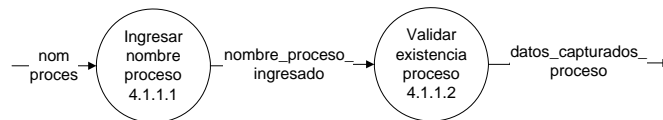


Figura N°8 Nivel 4.1.1, Ingresar Datos Proceso.

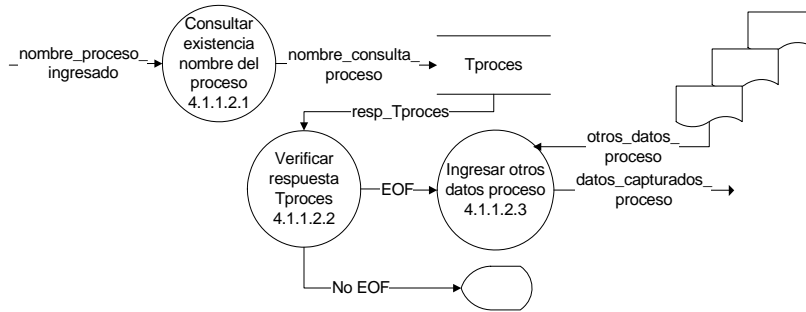


Figura N°9 Nivel 4.1.1.2, Validar Existencia de Proceso.

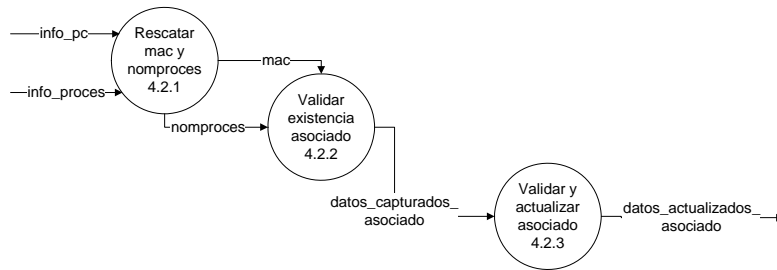


Figura N°10 Nivel 4.2, Asociar Procesos Validos.

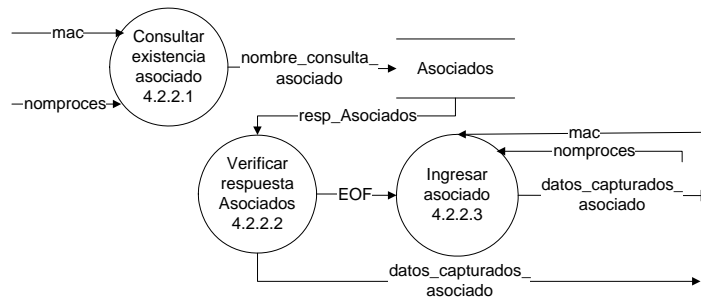


Figura N°11 Nivel 4.2.2, Validar Existencia Asociación.

6.3.1.3 Almacenes de Datos Utilizados por el Subsistema Procesos

Tproces	@nomproces+nameprogm+pathproccess+tamano
	Tabla en la que se encuentra la información de los procesos que se han identificado como permitidos
Nomproces	= 1{[A..Z][a..z][0..9]}255
	Código que identifica el nombre del proceso permitido
Nameprogrm	= 1{[A..Z][a..z][0..9]}255
	Nombre de la aplicación
Tamano	= 1{[0..9]}10
	<ul style="list-style-type: none">• Tamaño de la aplicación que genera el proceso*
Flujo de entrada	: datos_actualizados_proceso nombre_consulta_proceso
Flujo de salida	: info_proces datos_procesos_validos resp_Tproces
Volumen	: Crecimiento anual 1% al 25%

Computador = @mac+ip+nombrep+obs

*Tabla en la que se almacenan los datos de los computadores administrados por el sistema *

Mac = 2{[A..F][a..f][0..9]}2+5{[:][+]+2{[A..F][a..f][0..9]}2}5

* Número unívoco que identifica las tarjetas de red con su proveedor*

Ip = 1{[1..255]}3+3{.+1{[0..255]}3}3

Dirección asignada a cada computador según el protocolo de Internet

Nombrep = 1{[A..Z][a..z][0..9]}255

Nombre lógico del computador

Obs = 1{[A..Z][a..z][0..9]}255

Característica que permita distinguir el equipo computacional

*Flujo de salida : info_pc

Volumen : Crecimiento anual 1% al 25%

Asociados = @mac+@nomproces

Tabla en la que se asocian los procesos con los computadores. A cada computador le corresponde un grupo de procesos determinado.

Mac = *Referirse a Computador.mac*

Nomproces = *Referirse a Tproces.nomproces*

*Flujo de entrada : datos_actualizados_asociado

nombre_consulta_asociado

datos_asociado

*Flujo de salida : resp_asociado

Volumen : Crecimiento anual 20% al 40%

6.3.1.4 Procesos Involucrados

subsistema procesos (4)

Módulo principal en el que se realiza la inserción y eliminación de los procesos permitidos; también se efectúa el almacenamiento de los procesos permitidos en cada computador

*Flujo de entrada : [nomproces+otros_datos_proceso+info_pc+
info_proces]*

*Flujo de salida : datos_actualizados_proceso
datos_actualizados_asociado*

ingresar datos proceso (4.1)

Proceso para la validación del ingreso y captura de los datos

Flujo de entrada : nomproces

Flujo de salida : datos_capturados_proceso

Validar existencia proceso (4.1.1)

Proceso para la validación de la existencia del registro asociado a la clave ingresada

Flujo de entrada : nombre_proceso_ingresado

Flujo de salida : datos_capturados_proceso

asociar procesos validos (4.2)

Proceso en el que se realiza el ingreso, eliminación y despliegue de asociaciones de procesos válidos con los computadores

Flujo de entrada : [info_pc+info_proces]

Flujo de salida : datos_actualizados_asociado

Validar existencia asociado (4.2.1)

Proceso para la validación de la existencia de asociación entre el proceso válido y el computador

Flujo de entrada : [mac+nomproces]

Flujo de salida : datos_capturados_asociado

Ingresar nombre proceso

Ingreso vía teclado de la información del nombre del proceso

Flujo de entrada : nomproces

Flujo de salida : nombre_proceso_ingresado

*Algoritmo:

Almacenar nomproces en variable*

consultar existencia nombre del proceso

Genera la consulta a la base de datos usando el nombre del proceso

Flujo de entrada : nombre_proceso_ingresado

Flujo de salida : nombre_consulta_proceso

*Algoritmo:

especifica_parametros()

envia_parametros()

BD genera respuesta

BD envía respuesta*

verificar respuesta Tproces

*Verifica el tipo de respuesta que envía la base de datos considerando respuesta vacía (el nombre del proceso no existe por tanto no ha sido ingresado) o la tupla correspondiente al nombre de consulta *

Flujo de entrada : resp_Tproces

Flujo de salida : [EOF | No EOF]

Fin de archivo o no fin de archivo

*Algoritmo:

si respuesta vacía entonces

 ingresar otros datos proceso

sino

 mostrar otros datos proceso*

ingresar otros datos proceso

Proceso condicionado por el proceso verificar respuesta Tproces, como el nombre del proceso no existe se pide el ingreso de la información de los otros datos del proceso

Flujo de entrada : [EOF+otros_datos_proceso]

Flujo de salida : datos_capturados_proceso

*Algoritmo:

almacenar datos en variables*

verificar datos capturados

Verifica que la captura tanto manual (ingreso de los datos a través del teclado) como automática (a partir de la base de datos) sea la correcta. Esto es evaluado por el usuario ya que es una verificación visual (faltas de ortografía, modificación de datos, etc.). La respuesta está condicionada a que si los datos mostrados en pantalla, a juicio del usuario, son correctos

Flujo de entrada : [datos_capturados_proceso+otros_datos_proceso]

Flujo de salida : datos_actualizados_proceso

*Algoritmo:

si datos_capturados_actualizados_proceso = true entonces

insertar datos_actualizados_proceso en la base de datos

sino

modifica datos_capturados_proceso

insertar datos_actualizados_proceso en la base de datos *

Rescatar mac y nomproces

Rescata la dirección mac del computador desde la tabla Computador y el nombre del proceso desde la tabla Tproces

Flujo de entrada : [info_pc+info_proces]

Flujo de salida : [mac+nomproces]

*Algoritmo:

especifica_parametros()

envia_parametros()

BD genera respuesta

BD envía respuesta

Almacenar mac en variable

Almacenar nom_proces en variable*

Validar y actualizar asociado

Verifica que la captura tanto manual (ingreso de los datos a través del teclado) como automática (a partir de la base de datos) sea la correcta. Esto es evaluado por el usuario ya que es una verificación visual (faltas de ortografía, modificación de datos, etc.). La respuesta está condicionada a que si los datos mostrados en pantalla, a juicio del usuario, están correctos

Flujo de entrada : datos_capturados_asociado

Flujo de salida : datos_actualizados_asociado

*Algoritmo:

si datos_capturados_actualizados_asociado = true entonces

insertar datos_actualizados_asociado en la base de datos

sino

modifica datos_capturados_asociado

insertar datos_actualizados_asociado en la base de datos *

consultar existencia asociado

Genera la consulta a la base de datos usando la mac del computador y el nombre del proceso

Flujo de entrada : [mac+nomproces]

Flujo de salida : nombre_consulta_asociado

*Algoritmo:

especifica_parametros()

envia_parametros()

BD genera respuesta

BD envía respuesta*

verificar respuesta Asociados

*Verifica el tipo de respuesta que envía la base de datos considerando respuesta vacía o la tupla correspondiente a la consulta *

Flujo de entrada : resp_Asociados

Flujo de salida : [EOF | datos_capturados_asociado]

*Algoritmo:

si respuesta vacía entonces

ingresar asociado

sino

mostrar datos_capturados_asociado*

ingresar asociación

Proceso condicionado por el proceso verificar respuesta Asociacion, como no existe asociación entre el proceso y el computador es posible asignarla

Flujo de entrada : [EOF + mac+nomproces]

Flujo de salida : datos_capturados_asociado

*Algoritmo:

almacenar datos en variables*

generar listado de procesos por PC

Genera el listado de procesos asignado a cada computador. Este listado es posible verlo por pantalla

Flujo de entrada : datos_asociado

Flujo de salida : listado_procesos_pc

*Algoritmo:

mostrar en pantalla listado de procesos validos del pc (*)

generar listado de procesos validos

Genera el listado procesos válidos. Este listado es posible verlo por pantalla

Flujo de entrada : datos_procesos_validos

Flujo de salida : listado_procesos_validos

*Algoritmo:

mostrar en pantalla listado de procesos validos (*)

6.3.1.5 *Flujos de Datos*

Nomproces = *Refiérase a Tproces.nomproces*

Dato generado a partir del nombre del proceso válido

*Proceso origen : rescatar mac y nomproces

*Proceso destino : subsistema de control de procesos de S.O. (4)

ingresar datos proceso (4.1)

ingresar nombre proceso

validar existencia asociado (4.2.1)

consultar existencia asociado

ingresar asociado*

nombre_proceso_ingresado = *Refiérase a Tproces.nomproces*

Dato ingresado por pantalla

Proceso origen : ingresar nombre proceso

*Proceso destino : validar existencia proceso (4.1.1)

consultar existencia nombre proceso*

nombre_consulta_proceso = *Refiérase a Tproces.nomproces*

*Dato entregado como variable a una consulta de base de datos *

Proceso origen : consultar existencia nombre proceso

resp_Tproces = [EOF | No EOF]

*Genera información de fin de archivo (consulta vacía) o información de consulta exitosa *

Proceso destino : verificar respuesta Tproces

EOF = *valor válido: true*

Indica que la consulta no genera resultado en la búsqueda

Proceso origen : verificar respuesta Tproces

Proceso destino : ingresar otros datos proceso

No EOF = nameprogm+pathprocces+tamano

Datos de la tupla que cumple con la consulta a la base de datos

Proceso origen : verificar repuesta Tproces

otros_datos_proceso = nameprog+pathprocess+tamano

*Datos ingresados por el usuario cuando la tupla no existe en la base de datos *

*Proceso destino : subsistema de control de procesos de S.O. (4)

Ingresar otros datos proceso*

datos_capturados_proceso = nomproces+ nameprog+pathprocess+
tamano

Datos mostrados en pantalla luego de su ingreso vía teclado o mediante consulta a la base de datos

*Proceso origen : ingresar datos proceso (1.1)

validar existencia proceso (4.1.1)

ingresar otros datos proceso*

Proceso destino : verificar datos capturados

datos_actualizados_proceso = nomproces+ nameprog+pathprocess+
tamano

Datos válidos que se usarán en la actualización de la base de datos

*Proceso origen : subsistema control de procesos de S.O. (4)

verificar datos capturados*

datos_procesos_validos = nomproces+ nameprog+pathprocess+

tamano

Datos de cada proceso que se usarán para generar una lista de los procesos válidos

Proceso destino : generar listado procesos validos

Listado_procesos_validos = 1{datos_procesos_validos}

*Informe de salida del subsistema que representa la lista de procesos válidos.

Este informe será mostrado por pantalla.*

Proceso origen : generar listado procesos validos

Info_pc = nombrepc+mac+ip+obs

Información almacenada en la tabla Computador

*Proceso destino : subsistema control de procesos S.O. (4)

asociar procesos válidos (4.2)

rescatar mac y nomproces*

Info_proces = nomproces+nameprogm+pathprocess+

tamano

Información almacenada en la tabla Tproces

*Proceso destino : subsistema control de procesos S.O. (4)

asociar procesos validos (4.2)

rescatar mac y nomproces*

Mac = *Refiérase a Computador.mac*

Dato generado a partir de la dirección Mac del computador

*Proceso origen : rescatar mac y nomproces

*Proceso destino : validar existencia asociado (4.2.1)

consultar existencia asociado

ingresar asociado*

datos_capturados_asociado = nomproces+ mac

Datos mostrados en pantalla luego de su ingreso vía teclado o mediante consulta a la base de datos

*Proceso origen : validar existencia asociado (4.2.1)

verificar respuesta Asociados

ingresar asociado*

Proceso destino : validar y actualizar asociado

nombre_consulta_asociacion = *[mac+nomproces]*

*Dato entregado como variable a una consulta de base de datos *

Proceso origen : consultar existencia asociado

resp_Asociados = [EOF | mac+nomproces]

*Genera información de fin de archivo (consulta vacía) o información de consulta exitosa *

Proceso destino : verificar respuesta Asociados

EOF = *valor válido: true*

Indica que la consulta no genera resultado en la búsqueda

Proceso origen : verificar respuesta Asociados

Proceso destino : ingresar asociado

datos_actualizados_asociado = Mac+nomproces

Datos válidos que se usarán en la actualización de la base de datos

*Proceso origen : subsistema control de procesos de S.O. (4)

asociar procesos validos (4.2)*

validar y actualizar asociado*

datos_asociado = Mac+nombrep +nomproces

Datos de cada asociación proceso de computador que se usarán para generar una lista de los procesos válidos de cada pc

Proceso destino : generar listado procesos por PC

Listado_procesos_pc = 1{datos_asociado}

Informe de salida del subsistema que representa la lista de procesos válidos por computador. Este informe será mostrado por pantalla.

Proceso origen : generar listado de procesos por PC

6.3.2 Modelo Entidad Relación

El modelo entidad relación se basa en una percepción de un mundo real que consiste en un conjunto de objetos básicos llamados entidades y relaciones. Este esquema representa la estructura lógica global de la base de datos.

Una entidad es un objeto que existe, es distinguible de otros objetos y que está representado por un conjunto de atributos. Para cada atributo hay un conjunto de valores permitidos llamados dominio de ese atributo; por ejemplo un atributo permitido podría ser todas las cadenas de texto de una determinada longitud o todos los enteros positivos.

Finalmente, una relación es una asociación entre varias entidades.

6.3.2.1 Identificación del Conjunto de Entidades

Analizando los diagramas de flujos de datos que permitieron diseñar el primer prototipo es posible identificar las siguientes entidades tentativas:

- Procesos.

“Conjunto de todos los procesos de sistema operativo permitidos”

- Computador personal.

“Conjunto de todos los computadores a administrar”

- Tipos privilegios.

“Conjunto de los privilegios por asignar a los usuarios”

- Usuario.

“Conjunto de todas las personas que administran el sistema”

Y dado que eventualmente un usuario podrá tener varios privilegios con sus correspondientes datos de ingreso (login y palabra clave) se incluye una entidad débil en el modelo.

- Perfil.

“Conjunto de datos de ingreso de las personas que administran el sistema”

6.3.2.2 *Identificación de Relaciones*

A partir de las entidades detectadas es posible listar el conjunto de pares para verificar si existe algún tipo de relación entre ellas.

Observación: Los pares destacados son relaciones potenciales.

- **Procesos-Computador Personal.**
- Procesos-Tipos privilegios.
- Procesos-Usuario.
- Procesos-Perfil
- Computador personal-Tipos privilegios.
- Computador personal-Usuario.
- Computador-Perfil.
- **Usuario-Tipos Privilegios.**
- **Usuario-Perfil.**
- **Tipo de Privilegios-Perfil.**

6.3.2.2.1 Relaciones Potenciales

- Relación Procesos-Computador Personal (M-N)

“Los procesos de sistema operativo se asocian a los computadores”

- Relación Usuario-Tipos privilegios (M-N)

“Los usuarios tienen tipos de privilegios”

- Relación Usuario-Perfil (1-N)

“El usuario posee perfiles”

- Relación Tipo privilegios-Perfil (1-N)

“Un tipo de privilegio define los perfiles”

En esta fase del diseño es posible, dada la definición de las entidades y sus relaciones potenciales, descartar la relación M-N Usuario-Tipos privilegios utilizando Usuario-Perfil y Tipo privilegios-Perfil ambas 1-N.

En esta etapa no se establece una relación entre las entidades Usuario y Computador personal, dado que la implementación de ésta no se requiere para cumplir con los objetivos planteados en el Seminario de Titulación y sin embargo, se incluirá en mejoras posteriores.

6.3.2.2.2 Diagrama Entidad-Relación Tentativo

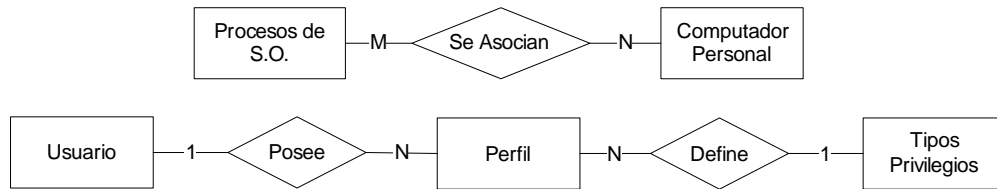


Figura N°12 Diagrama Entidad Relación Tentativo.

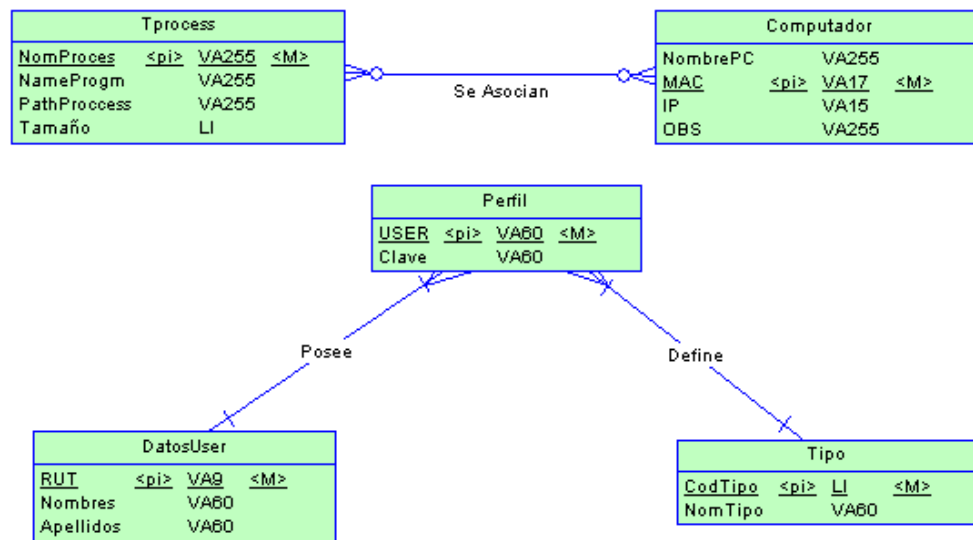


Figura N°13 Modelo Conceptual Según PowerDesigner Trial 9

El diagrama, cuya descripción de simbología se muestra en el anexo 11.4, presenta relaciones M-N que deben ser transformadas.

6.3.2.3 Refinación del Modelo de Datos

La refinación implica la eliminación de relaciones M-N. A partir de esta refinación se obtendrán nuevas entidades o entidades débiles en que sus claves primarias dependerán de las entidades padre.

6.3.2.3.1 Refinación del Modelo Entidad Relación

- Relación “Los computadores ejecutan procesos de sistema operativo”

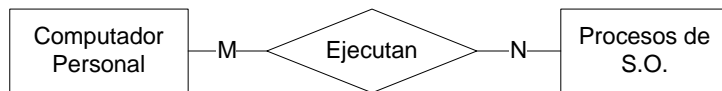


Figura Nº14 Relación “Los Computadores Ejecutan Procesos de Sistema Operativo”.

Esta relación se convierte en:

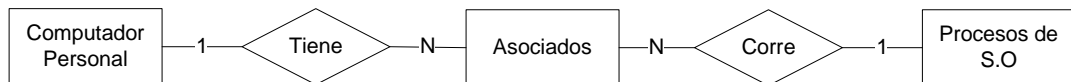


Figura Nº15: Refinación “Los Computadores Ejecutan Procesos de Sistema Operativo”.

“El computador personal tiene procesos asociados” (1-N)

“Un proceso de sistema operativo corre en varias máquinas asociadas” (1-N)

6.3.2.3.2 *Conjunto de Entidades Padre*

- Procesos.
- Computador personal.
- Tipos privilegios.
- Usuario.

6.3.2.3.3 *Conjunto de Entidades Hijo*

- Perfil.
- Asociados.

6.3.2.3.4 *Relación entre Entidades*

- Relación Computador personal-Asociado (1-N)

“El computador personal tiene procesos asociados”

- Relación Procesos-Asociado (1-N)

“Un proceso de sistema operativo corre en varias máquinas asociadas”

- Relación Usuario-Perfil (1-N)

“El usuario posee perfiles” (1-N)

- Relación Tipo privilegio-Perfil (1-N)

“Un tipo de privilegio define los perfiles” (1-N)

6.3.2.3.5 Normalización del Modelo

Este proceso permite decidir qué atributos pertenecen a una entidad, lo que asegura que el modelo sea estable (no pierde información), con un mínimo de redundancia (repita datos) y sea capaz de representar toda la información.

Revisando los atributos definidos para cada entidad y comparando con las entradas del sistema es posible afirmar que el modelo es capaz de representar toda la información requerida.

El aumento de la cantidad de entidades hijo permite eliminar los datos repetidos pues originalmente, para poder responder a distintas consultas, en las tablas se repetían los atributos.

La descomposición anterior si bien, ayuda a evitar la redundancia, puede provocar que una consulta no entregue todos los datos con la consecuente pérdida de información. Sin embargo, luego de probar las salidas requeridas se encuentra ante un modelo robusto que responde a las transacciones en forma íntegra. Así, luego de analizar los grupos de atributos en cada relación es posible afirmar que el modelo cumple la primera forma normal, pues no existen grupos repitentes, cumple la segunda forma normal, ya que los atributos que no son claves primarias dependen completamente de ella, cumple la tercera forma normal ya que no se presentan dependencias transitivas de la clave primaria y finalmente, cumple la forma normal BCNF,

ya que cada uno de los miembros de los esquemas de relación que comprenden el diseño están en BCNF y los determinantes de cada dependencia funcional (lado izquierdo) son claves candidatas.

6.3.2.4 Identificación de los Atributos Existentes

- Entidad Procesos.

Tproces	= nomproces+nameprog+pathprocess+tamano
nom_proces	= Código que identifica el nombre del proceso permitido Alfanumérico (255) Valor nulo:falso
Nameprog	= Nombre de la aplicación Alfanumérico (255) Valor nulo:falso
Pathprocess	= Nombre del régimen establecido Alfanumérico (255) Valor nulo:falso
Tamano	= Tamaño de la aplicación que genera el proceso Numérico Valor nulo:falso

- Entidad Computador personal.

Computador = mac+ip+nombrep+obs

Mac = Número unívoco que identifica las tarjetas de red con su proveedor

Alfanumérico (17)

Valor nulo:falso

Valores permitidos: [A..F] [0..9] [:-]

Ip = Dirección asignada a cada computador según el protocolo de Internet

Alfanumérico (15)

Valor nulo:falso

Valores permitidos: [0..255] [.]

Nombrep+ = Nombre lógico del computador

Alfanumérico (255)

Valor nulo:falso

Obs = Característica que permita distinguir al equipo computacional

Alfanumérico (255)

Valor nulo:verdadero

- Entidad Tipos privilegios.

Tipo = codtipo+nomtipo

Codtipo = Código para el tipo de privilegio

Numérico

Valor nulo:falso

Nomtipo = Nombre para el tipo de privilegio

Alfanumérico (60)

Valor nulo:falso

- Entidad Asociados.

Asociados = nomproces+mac

Nomproces = Número unívoco que identifica las tarjetas de red con su proveedor

Alfanumérico (17)

Valor nulo:falso

Valores permitidos: [A..F] [a..f] [0..9] [:-]

Mac = Código que identifica el nombre del proceso permitido

Alfanumérico (255)

Valor nulo:falso

- Entidad Usuario.

Datosuser = rut+nombres+apellidos

Rut = RUT del usuario que accede a la aplicación

Alfanumérico (9)

Valor nulo:falso

Valores permitidos: [0..9] [k][K]

Nombres = Nombres del usuario que accede a la aplicación

Alfanumérico (60)

Valor nulo:falso

Apellidos = Apellidos del usuario que accede a la aplicación

Alfanumérico (60)

Valor nulo:falso

- Entidad Perfil.

Perfil = user+rut+codtipo+clave

User = Alias ingresado para autenticación del usuario

Alfanumérico (60)

Valor nulo:falso

Rut = RUT del usuario que accede a la aplicación

Alfanumérico (9)

Valor nulo:falso

Valores permitidos: [0..9] [k] [K]

Codtipo = Código para el tipo de privilegio

Numérico

Valor nulo:falso

Clave = Palabra clave ingresada para la autenticación del usuario

Alfanumérico (60)

Valor nulo:falso

6.3.2.5 Identificación del Dominio de los Atributos

nom_proces = Alfanumérico

Tamaño 255

Ejemplo: AcroRd32.exe

Nameprogm = Alfanumérico

Tamaño 255

Ejemplo: Acrobat Reader

Pathprocess = Alfanumérico

Tamaño 255

Ejemplo: C:\ Archivos de programa\ Adobe\

Acrobat 6.0\ Reader

Tamano = Numérico

Tamaño [0-2147483647]

Ejemplo: 7493

Mac = Alfanumérico
Tamaño 17
Ejemplo: 00-01-4ª-F2-46-69

Ip = Alfanumérico
Tamaño 15
Ejemplo: 192.1.1.10

Nombrep = Alfanumérico
Tamaño 255
Ejemplo: pc5

Obs = Alfanumérico
Tamaño 255
Ejemplo: Director

Codtipo = Numérico
Tamaño [0-2147483647]
Ejemplo: 1

Nomtipo = Alfanumérico

Tamaño 60

Ejemplo: Administrador

Rut = Alfanumérico

Tamaño 9

Ejemplo: 134051744

Nombres = Alfanumérico

Tamaño 60

Ejemplo: Aris David

Apellidos = Alfanumérico

Tamaño 60

Ejemplo: Iturra Soto

User = Alfanumérico

Tamaño 60

Ejemplo: aiturra

Clave = Alfanumérico

Tamaño 60

Ejemplo: a5yk3

6.3.2.6 Identificación de las Claves Primarias

Tproces = Nomproces

Computador = Mac

Tipo = Codtipo

Datosuser = Rut

Asociados = nomproces+mac

Perfil = user+rut

6.3.2.7 Diagrama Entidad Relación Final

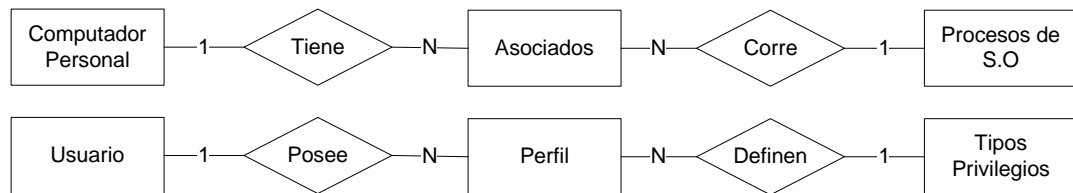


Figura N°16 Diagrama Entidad Relación Final

6.3.2.8 Transacciones

Las transacciones están definidas por los listados que se desean mostrar en pantalla.

Transacción T(1)

- “Obtener el listado de todos los procesos válidos“

Transacción T(2)

- “Obtener el listado de los computadores personales administrados por el sistema”

Transacción T(3)

- “Obtener el listado de los usuarios permitidos por el sistema”

Transacción T(4)

- “Obtener el listado de los procesos válidos por computador”

Transacción T(5)

- “Autenticación de usuario”

6.3.2.9 Validación del Modelo con las Transacciones

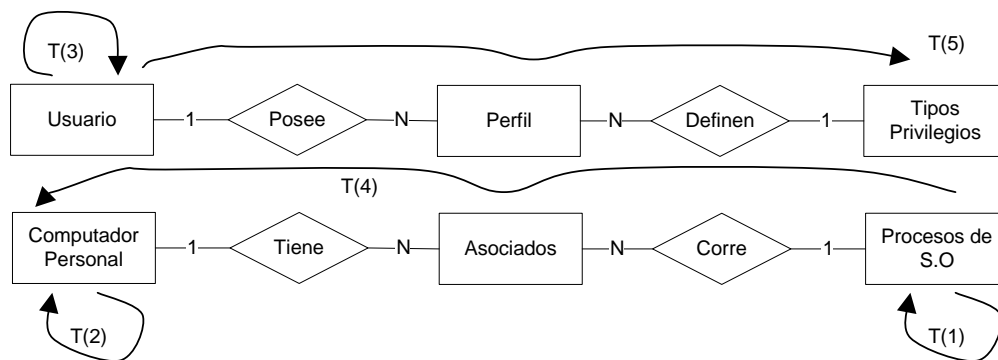


Figura N°17 Validación de las transacciones en el diagrama Entidad Relación

6.3.2.10 Definición de Limitaciones de Integridad Referencial y Derivación de Relaciones

- Procesos

Tproces(nomproces,nameprogm,pathproccess,tamano)

primary key nomproces

- Computador personal.

Computador(mac,ip,nombrepc,obs)

primary key mac

- Tipos privilegios.

Tipo(codtipo,nomtipo)

primary key codtipo

- Usuario.

Datosuser(rut,nombres,apellidos)

primary key rut

- Asociados.

Asociados(nomproces,mac)

primary key nomproces,mac

Foreign key nomproces references Tproces

on delete cascade on update cascade

Foreign key mac references Computador

on delete cascade on update cascade

- Perfil.

Perfil(user,rut,codtipo,clave)

primary key user

Foreign key rut references Datosuser

on delete cascade on update cascade

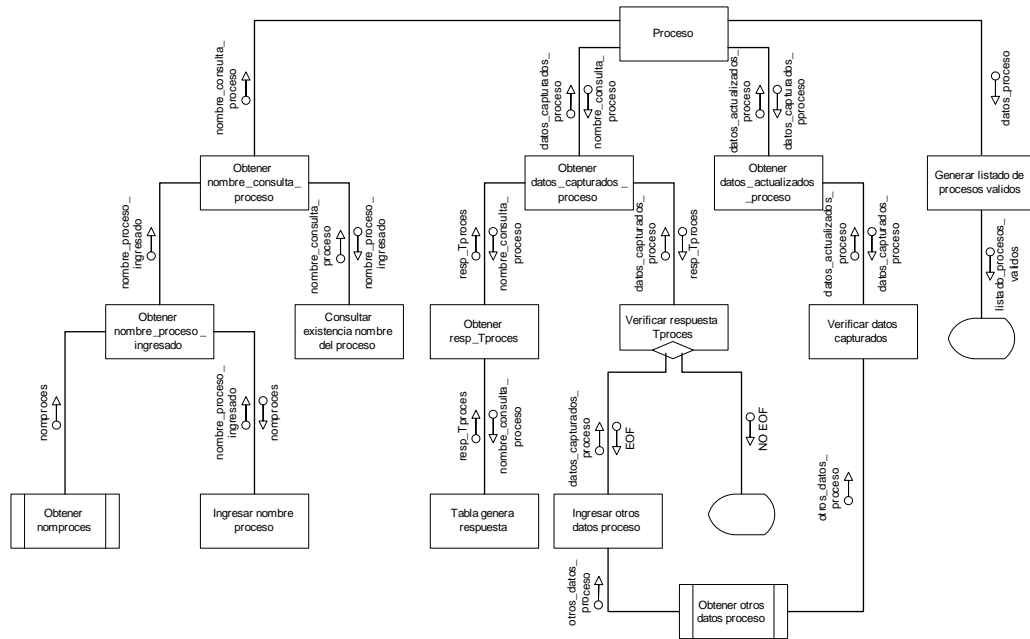
Foreign key codtipo references Tipo

on delete cascade on update cascade

6.3.3 Diseño Estructurado

En el diseño estructurado se convierten los diagramas de flujos de datos a diagramas de estructura en que cada proceso del DFD pasa a ser un módulo de programa con su correspondiente especificación de código.

6.3.3.1 Diagrama Estructura Subsistema Control de Procesos de Sistema Operativo



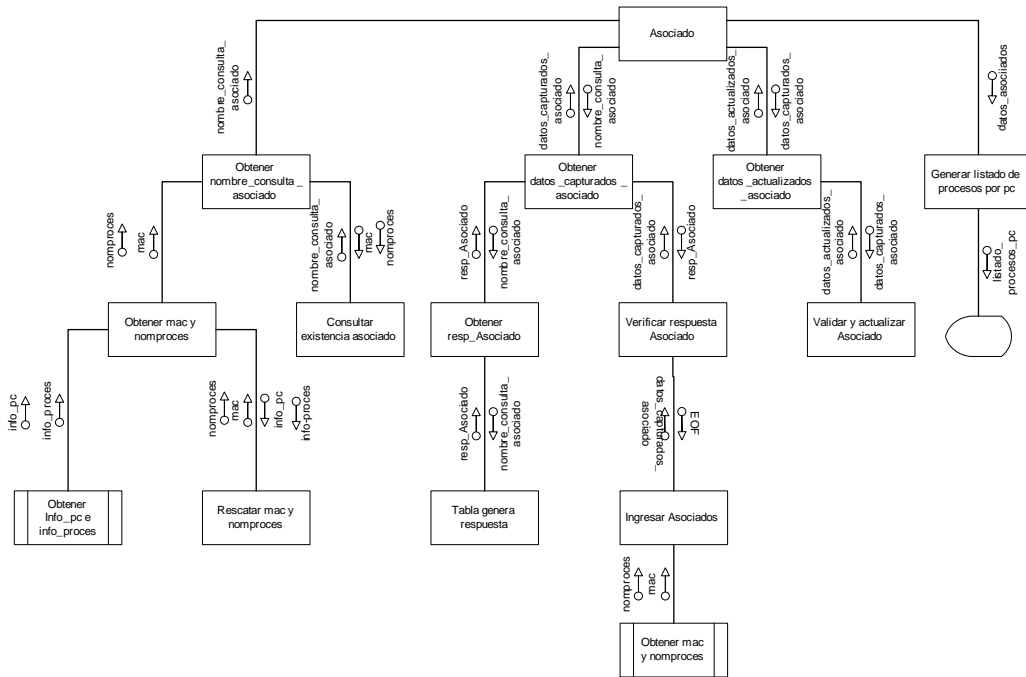


Figura N°18 Diagrama de estructura Subsistema Control de Procesos de S.O.

6.3.3.2 Especificación de Módulos Subsistema Control de Procesos de Sistema Operativo

- Módulo : Obtener nomproces
- Usa : Nomproces
- Función : comprueba validez de nomproces
- Retorna : Nomproces

- Módulo : Ingresar nomproces
- Usa : Nomproces
- Función : almacena nomproces en variable

Retorna : Nombre_proceso_ingresado

Módulo : Obtener nombre_proceso_ingresado

Usa : Nombre_proceso_ingresado

Función : almacena nombre_proceso_ingresado en variable

Retorna : Nombre_proceso_ingresado

Módulo : Consultar existencia nombre del proceso

Usa : Nombre_proceso_ingresado

Función : crea_consulta (nombre_proceso_ingresado)

Retorna : Nombre_consulta_proceso

Módulo : Obtener nombre_consulta_proceso

Usa : Nombre_consulta_proceso

Función : almacena consulta en variable

Retorna : Nombre_consulta_proceso

Módulo : Tabla genera respuesta

Usa : Nombre_consulta_proceso

Función : tabla genera respuesta

Retorna : resp_Tproces

Módulo : Obtener resp_Tproces

Usa : resp_Tproces

Función : almacena resp_Tproces en variables

Retorna : resp_Tproces

Módulo : Obtener otros_datos_proceso

Usa : otros_datos_proceso

Función : almacena otros_datos_proceso en variable

Retorna : otros_datos_proceso

Módulo : Verificar respuesta Tproces

Usa : resp_Tproces

Función : verifica respuesta (resp_Tproces)

Retorna : EoF/NO EoF

Módulo : Ingresar otros datos proceso

Usa : EoF, otros_datos_proceso

Función : almacenar registro en variable

almacenar otros_datos_proceso en variables

Retorna : datos_capturados_proceso

Módulo : Obtener datos_capturados_proceso

Usa : datos_capturados_proceso
 Función : almacena datos_capturados_proceso en variables
 Retorna : datos_capturados_proceso

Módulo : Verificar datos capturados
 Usa : datos_capturados_proceso
 Función : retorna datos_actualizados_proceso
 Retorna : datos_actualizados_proceso

Módulo : Obtener datos_actualizados_proceso
 Usa : datos_actualizados_proceso
 Función : almacena datos actualizados_proceso en variables
 Retorna : datos_actualizados_proceso

Módulo : Generar listado procesos validos
 Usa : datos_procesos_validos
 Función : crea consulta de selección a la tabla

 tabla envia respuesta

 muestra en pantalla listado procesos validos

Retorna : listado_procesos_validos
 Módulo : Obtener info_pc, info_proces
 Usa : info_pc/info_proces

Función : comprueba validez de info_pc e info_proces

Retorna : info_pc/info_proces

Módulo : Rescatar mac y nomproces

Usa : info_pc/info_proces

Función : almacena mac y nomproces

Retorna : mac/nomproces

Módulo : Obtener mac y nomproces

Usa : mac/nomproces

Función : almacena mac y nomproces en variable

Retorna : mac/nomproces

Módulo : Consultar existencia asociado

Usa : mac/nomproces

Función : crea_consulta (mac,nomproces)

Retorna : Nombre_consulta_asociado

Módulo : Obtener nombre_consulta_asociado

Usa : Nombre_consulta_asociado

Función : almacena consulta en variable

Retorna : Nombre_consulta_asociado

Módulo : Tabla genera respuesta

Usa : Nombre_consulta_asociado

Función : tabla genera respuesta

Retorna : resp_Asociado

Módulo : Obtener resp_Asociado

Usa : resp_Asociado

Función : almacena resp_Asociado en variables

Retorna : resp_Asociado

Módulo : Verificar respuesta Asociado

Usa : resp_Asociado

Función : verifica respuesta (resp_Asociado)

Retorna : EoF,datos_capturados_asociado

Módulo : Ingresar asociados

Usa : EoF, mac, nomproces

Función : almacenar registro en variable

almacenar mac, nomproces en variables

Retorna : Datos_capturados_asociado

Módulo : Obtener datos_capturados_asociado

Usa : Datos_capturados_asociado

Función : almacena datos_capturados_asociado en variables

Retorna : Datos_capturados_asociado

Módulo : Validar y actualizar asociado

Usa : Datos_capturados_asociado

Función : retorna datos_actualizados_asociado

Retorna : Datos_actualizados_asociado

Módulo : Obtener datos_actualizados_asociado

Usa : Datos_actualizados_asociado

Función : almacena datos actualizados_asociado en variables

Retorna : Datos_actualizados_asociado

Módulo : Generar listado de procesos por pc

Usa : Datos_asociados

Función : crea consulta de selección a la tabla

tabla envia respuesta

muestra en pantalla listado procesos por pc

Retorna : Listado_procesos_pc

6.3.4 Modelo Físico y Detalle de la Base de Datos

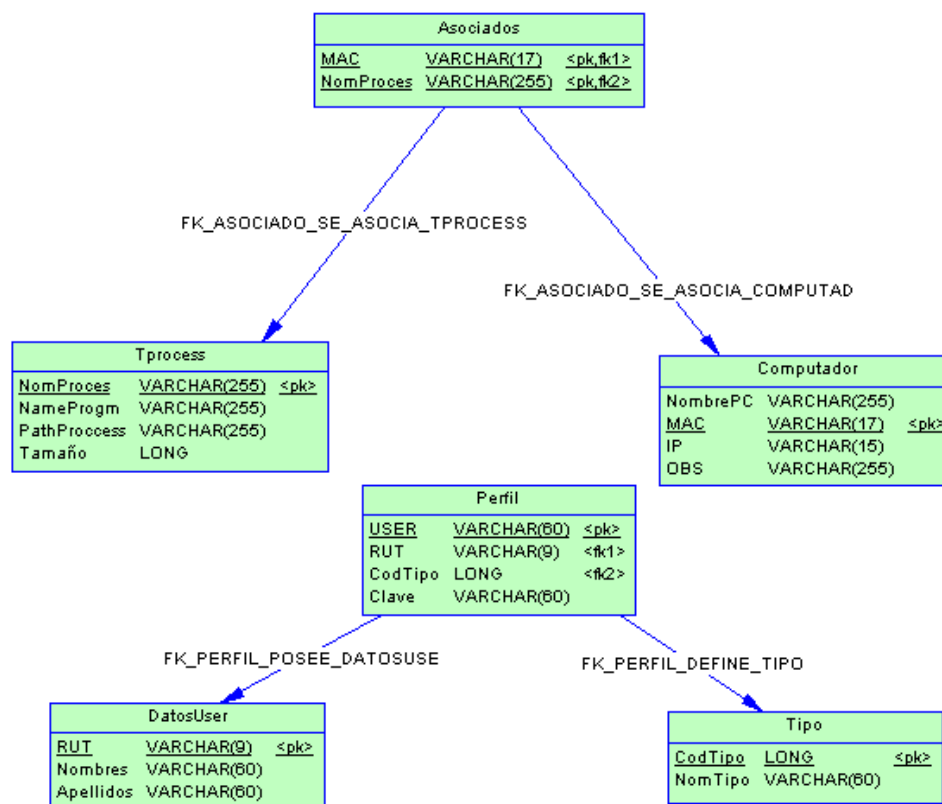


Figura N°19 Modelo Físico del sistema. según PowerDesigner Trial 9

- **Tprocess**(NomProces,NameProgram,PathProcecess,Tamaño)

Esta tabla se encarga de los procesos que se han identificado como permitidos. Contiene los siguientes datos: nombre del proceso (NomProceso) como clave primaria; el nombre de la aplicación, (NameProgram), ubicación en el disco (*PathProcess*) y tamaño (*Tamaño*) de la aplicación que genera el proceso.

	Name	Code	Data Type	P	F	M
→	NomProces	NOMPROCES	VARCHAR(255)	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
2	NameProgm	NAMEPROGM	VARCHAR(255)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
3	PathProcess	PATHPROCCES	VARCHAR(255)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
4	Tamaño	TAMANO	LONG	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Figura N°20 Tipo de datos de la tabla Tproces.

- **Computador**(NombrePC,MAC,IP,OBS)

Esta tabla se encarga de almacenar los datos de los computadores administrados por el sistema. La descripción de sus campos es la siguiente: Clave primaria MAC es un número unívoco que identifica las tarjetas de red con su proveedor, dado que hoy en día la gran mayoría vienen integradas al computador. Este número permite identificarlo.

Contiene además los campos NombrePC el cual define el nombre del computador, IP dirección de Internet en el cual de puede ubicar el computador, OBS campo que contiene observaciones Ej: perteneciente al gerente.

	Name	Code	Data Type	P	F	M
1	NombrePC	NOMBREPC	VARCHAR(255)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
2	MAC	MAC	VARCHAR(17)	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
3	IP	IP	VARCHAR(15)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
4	OBS	OBS	VARCHAR(255)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Figura N°21 Tipo de datos de la tabla Computador.

- **Asociados**(NomProces,MAC)

En esta tabla se asocia los procesos con los computadores a cada computador le corresponde un grupo de procesos determinados.

Para esta tabla existen dos claves primarias

- NomProces : nombre de los procesos.
- MAC: a través de este código se identifica el computador.

	Name	Code	Data Type	P	F	M
1	MAC	MAC	VARCHAR(17)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
2	NomProces	NOMPROCES	VARCHAR(255)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Figura N°22 Tipo de datos de la tabla Asociados.

- **Perfil**(USER,RUT,CodTipo,Clave)

Contiene los usuarios que pueden que pueden acceder a la aplicación.

	Name	Code	Data Type	P	F	M
1	USER	USER	VARCHAR(60)	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
2	RUT	RUT	VARCHAR(9)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
3	CodTipo	CODTIPO	LONG	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
4	Clave	CLAVE	VARCHAR(60)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Figura N°23 Tipo de datos de la tabla Perfil.

- **DatosUser**(RUT, Nombres, Apellidos)

Contiene los datos de los usuarios que pueden acceder a la aplicación

	Name	Code	Data Type	P	F	M
1	RUT	RUT	VARCHAR(9)	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
2	Nombres	NOMBRES	VARCHAR(60)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
3	Apellidos	APELLIDOS	VARCHAR(60)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Figura N°24 Tipo de datos de la tabla Características.

- **Tipo**(CodTipo, NomTipo)

Contiene los tipos de usuario que existen para la aplicación

	Name	Code	Data Type	P	F	M
→	CodTipo	= CODTIPO	LONG	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
2	NomTipo	NOMTIPO	VARCHAR(60)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Figura N°25 Tipo de datos de la tabla Tipo.

7. IMPLEMENTACIÓN PERFECCIONAMIENTO DEL PROTOTIPO FUNCIONAL Y POSTERIOR ANÁLISIS DE LA SITUACIÓN DEL ENTORNO OPERACIONAL DE LOS PCS, UNA VEZ INSTALADO EL PROTOTIPO FUNCIONAL

7.1 Desarrollo

7.1.1 Codificación y Diseño de Interfaz

7.1.1.1 Código

- *Llamada a librerías externas*

Muchas de las funciones con las que opera el prototipo, corresponden a llamadas de sistema, y el modo de interactuar con estas es a través de librerías dinámicas (DLLs), las cuales son accedidas a través de llamadas a la API (interfaz de programación de aplicaciones)

```
Option Explicit
'*****
'Declaracion de apis de Apagado de equipo
'*****
Public Declare Function ExitWindowsEx& Lib "user32" (ByVal uFlags&, ByVal dwReserved&)
'*****
'Declaracion de apis de manejo de proceso
'*****
Public Declare Function GetCurrentProcessId Lib "kernel32" () As Long
'*****
'Dim result As Integer
'*****
Public Declare Function GetAsyncKeyState Lib "user32" (ByVal vKey As Long) As Integer
```

- *Monitoreo de Procesos*

Procedimiento que se encarga de capturar los nombres y atributos de los procesos que se gestan en el sistema operativo. También se encarga de monitorearlos y eliminarlos.

```

Private Sub Timer1_Timer()
Dim Num As Integer
Dim cad As String
Dim NameProcess As String
DoEvents
ListKillProcess1.List_Processes = True
ListKillProcess1.Get_Process_Info = MaxNumberOfProcesses 'ListKillProcess1
If ListKillProcess1.Number_Of_Processes > MaxNumberOfProcesses Then
ListKillProcess1.Get_Process_Info = ListKillProcess1.Number_Of_Processes
Num = InStr(ListKillProcess1.Process_Name, Chr(0)) - 1
If Num <> -1 Then
NameProcess = Mid(ListKillProcess1.Process_Name, 1, Num)
Else
NameProcess = ""
End If
If ListKillForm.Visible = False Then
If VerifProces("CProcess", NameProcess) = True Then
cad = ListKillProcess1.Process_Name
Kill_Progm
Else
MaxNumberOfProcesses = ListKillProcess1.Number_Of_Processes
End If
Else
If VerifProces("CProcess", NameProcess) = True Then
Crga_Procces MSFlexGrid1, ListKillProcess1.Process_Name
MaxNumberOfProcesses = ListKillProcess1.Number_Of_Processes
End If
End If
Else
MaxNumberOfProcesses = ListKillProcess1.Number_Of_Processes
End If
If -32767 = GetAsyncKeyState(17) Then
If -32767 = GetAsyncKeyState(18) Then
If -32767 = GetAsyncKeyState(86) Then
FrmPassWord.Show
End If
End If
End If
End Sub

```

- *Asesino de Procesos*

Procedimiento que se encarga de eliminar todos aquellos procesos que no concuerden con la lista de atributos que se encuentra en la base de datos. Los atributos de procesos que se encuentran en la base de datos corresponden a los procesos válidos

```

Public Sub KillProces()
Dim Num As Integer
Dim i As Integer
Dim cad As String
Dim NameProcess As String
ListKillProcess1.List_Processes = True
MSFlexGrid1.Clear
MSFlexGrid1.Rows = 0
For i = 1 To ListKillProcess1.Number_Of_Processes
ListKillProcess1.Get_Process_Info = i
Num = InStr(ListKillProcess1.Process_Name, Chr(0)) - 1
If Num <> -1 Then
NameProcess = Mid(ListKillProcess1.Process_Name, 1, Num)
Else
NameProcess = ""
End If
If VerifProces("CProcess", NameProcess) = True Then
cad = ListKillProcess1.Process_Name
Kill_Progm
End If
Next i
MaxNumberOfProcesses = ListKillProcess1.Number_Of_Processes
End Sub

```

- *Despliega Procesos*

Permite mostrar por pantalla los nuevos procesos que se están ejecutando en el sistema operativo y que no están dentro de los procesos válidos. Este procedimiento sólo funciona si la aplicación está en modo administración.

```
Public Sub ShowProcess()  
Dim Num As Integer  
Dim i As Integer  
Dim NameProcess As String  
ListKillProcess1.List_Processes = True  
MSFlexGrid1.Clear  
MSFlexGrid1.Rows = 0  
For i = 1 To ListKillProcess1.Number_Of_Processes  
ListKillProcess1.Get_Process_Info = i  
Num = InStr(ListKillProcess1.Process_Name, Chr(0)) - 1  
If Num <> -1 Then  
NameProcess = Mid(ListKillProcess1.Process_Name, 1, Num)  
Else  
NameProcess = ""  
End If  
If VerifProces("CProcess", NameProcess) = True Then  
Crga_Procces MSFlexGrid1, ListKillProcess1.Process_Name  
End If  
Next i  
MaxNumberOfProcesses = ListKillProcess1.Number_Of_Processes  
End Sub
```

- *Verificar existencia de usuario*

Función que permite validar la existencia de un usuario, esta función pertenece al submódulo de seguridad de el prototipo.

```
'Verifica la existencia del usuario y la clave
Public Function VrifUser(USER As String, Pswrd As String) As Integer
    Dim dbs As Database
    Dim Qdf As QueryDef
    Dim Rst As Recordset
    Dim prm As Parameter
    ' Devuelve la referencia a la base de datos activa.
    IsFileUnLocked Path, 5
    Set dbs = Workspaces(0).OpenDatabase(Path, True, False, ";pwd=Enkidu")
    Set Qdf = dbs.QueryDefs("CDtsCtaUsr")
    ' Suministra valores a los parámetros.
    Qdf.Parameters![USER] = USER
    Qdf.Parameters![Pswrd] = Pswrd
    ' Abre como Recordset un objeto QueryDef.
    Set Rst = Qdf.OpenRecordset
    If Rst.EOF = False Then
        VrifUser = Rst.Fields("CODTIPO")
    Else
        VrifUser = 0
    End If
    Qdf.Close
    Rst.Close
    dbs.Close
    IsFileLocked Path, 5
End Function
```

- *Guarda proceso*

Procedimiento que almacena en la base de datos, los atributos correspondientes a los procesos válidos.

```

Public Sub GrdaProceso(NameProgram As String, NameProces As String)
    'CambiaBloques (Path)
    IsFileUnLocked Path, 5
    Dim dbs As Database
    Dim Qdf As QueryDef
    Set dbs = Workspaces(0).OpenDatabase(Path, True, False, ";pwd=Enkidu")
    Set Qdf = dbs.QueryDefs("IProcesos")
    Qdf.Parameters![NameProgram] = NameProgram
    Qdf.Parameters![Proces] = NameProces
    On Error GoTo Err_Ejecutar
    Qdf.Execute dbFailOnError
    On Error GoTo 0
    Qdf.Close
    dbs.Close
    IsFileLocked Path, 5
    'CambiaBloques (Path)
    Exit Sub
Err_Ejecutar:
    If DBEngine.Errors.Count > 0 Then
        For Each errBucle In DBEngine.Errors
            MsgBox "Número de error: " & errBucle.Number & vbCrLf & _
                errBucle.Description
        Next errBucle
    End If
Resume Next
End Sub

```


- *Elimina procesos*

Procedimiento que permite eliminar de la base de datos, los atributos de un proceso designado como válido

```
Public Sub EliminaProceso(NameProcess As String)
Dim dbs As Database
Dim Qdf As QueryDef
'CambiaBloques (Path)
IsFileUnLocked Path, 5
Set dbs = Workspaces(0).OpenDatabase(Path, True, False, ";pwd=Enkidu")
Set Qdf = dbs.QueryDefs("DeleteProcess")
Qdf.Parameters![Proces] = NameProcess
On Error GoTo Err_Ejecutar
Qdf.Execute dbFailOnError
On Error GoTo 0
Qdf.Close
dbs.Close
'CambiaBloques (Path)
IsFileLocked Path, 5
Exit Sub
Err_Ejecutar:
If DBEngine.Errors.Count > 0 Then
For Each errBucle In DBEngine.Errors
MsgBox "Número de error: " & errBucle.Number & vbCr & _
errBucle.Description
Next errBucle
End If
Resume Next
End Sub
```

- *Verificar existencia de procesos válidos*

Función que verifica la existencia del nombre de un proceso en la base de datos

```
Public Function VerifProces(Tbla As String, Dtos As String) As Boolean
    Dim dbs As Database
    Dim Qdf As QueryDef
    Dim Rst As Recordset
    Dim prm As Parameter
    'CambiaBloques (Path)
    IsFileUnLocked Path, 5
    ' Devuelve la referencia a la base de datos activa.
    Set dbs = Workspaces(0).OpenDatabase(Path, True, False, ";pwd=Enkidu")
    Set Qdf = dbs.QueryDefs(Tbla)
    ' Suministra valores a los parámetros.
    Qdf.Parameters![Proces] = Dtos
    CnsltadTos = True
    ' Abre como Recordset un objeto QueryDef.
    Set Rst = Qdf.OpenRecordset
    If Rst.EOF = False Then
        VerifProces = False
    Else
        VerifProces = True
    End If
    Qdf.Close
    Rst.Close
    dbs.Close
    'CambiaBloques (Path)
    IsFileLocked Path, 5
End Function
```

- Lista Procesos

Procedimiento que permite listar, desde la base de datos, los procesos, que hallan sido designados como válidos. Se listan además los atributos asociados a éstos.

```
Public Sub CrgProces(MsFlexGrd As MSFlexGrid)
Dim Db As Database
Dim Qdf As QueryDef
Dim Rst As Recordset
Dim NombrePross As String
Dim NombreProGrm As String
'CambiaBloques (Path)
IsFileUnLocked Path, 5
Set Db = Workspaces(0).OpenDatabase(Path, True, False, ";pwd=Enkidu")
Set Qdf = Db.QueryDefs("CPctaPcrocoss")
Set Rst = Qdf.OpenRecordset
With Rst
Do While (.EOF = False)
.Edit
NombrePross = Rst.Fields("NomProces")
NombreProGrm = Rst.Fields("NameProgm")
Crga_Flexgr MsFlexGrd, NombrePross, NombreProGrm

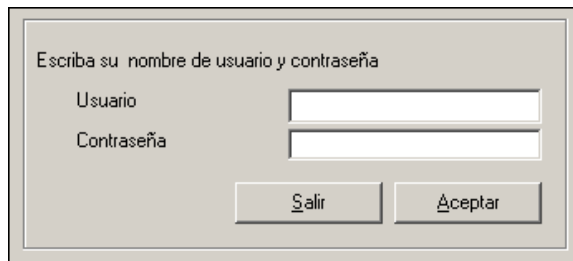
.MoveNext
Loop
End With
Rst.Close
Db.Close
'CambiaBloques (Path)
IsFileLocked Path, 5
End Sub
```

7.1.1.2 *Diseño de Interfaces*

El diseño de interfaces es una aproximación real, de las pantallas a utilizar en el sistema.

7.1.1.2.1 *Diseño de Interfaz Inicio de Sesión*

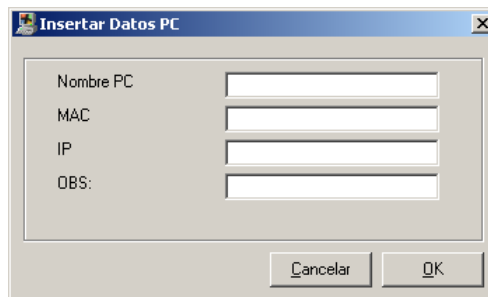
El Diseño de interfaz del cuadro de diálogo inicio de sesión cuenta con dos campos de entrada de texto donde se solicitan el nombre de usuario y la clave, para luego ser enviados por el botón aceptar a un proceso de validación.



Es un cuadro de diálogo con un título que dice "Escriba su nombre de usuario y contraseña". Contiene dos campos de entrada de texto etiquetados como "Usuario" y "Contraseña". Debajo de los campos hay dos botones: "Salir" y "Aceptar".

Figura N°26 Diseño de Interfaz de Inicio de Sesión

7.1.1.2.2 *Diseño de Inserción y Eliminación de Equipos.*



Es un cuadro de diálogo con un título que dice "Insertar Datos PC". Contiene cuatro campos de entrada de texto etiquetados como "Nombre PC", "MAC", "IP" y "OBS:". Debajo de los campos hay dos botones: "Cancelar" y "OK".

Figura N°27 Diseño de Interfaz de Equipos.

7.1.1.2.3 Diseño de Interfaz Inserción Procesos y Asociación con Aplicación y Equipos

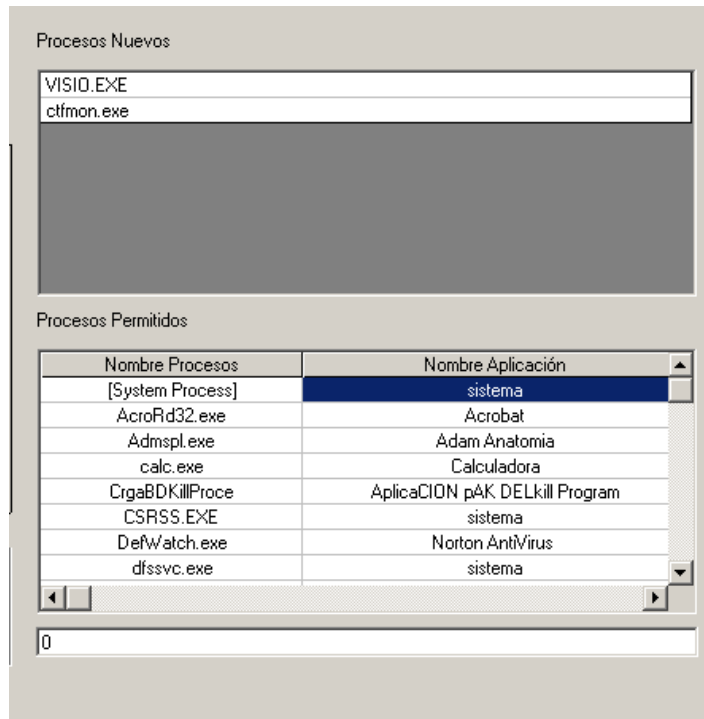


Figura N°28 Diseño de Interfaz de Procesos y Aplicaciones y Equipos

7.1.1.2.4 Diseño de Inserción y Eliminación de Usuarios.

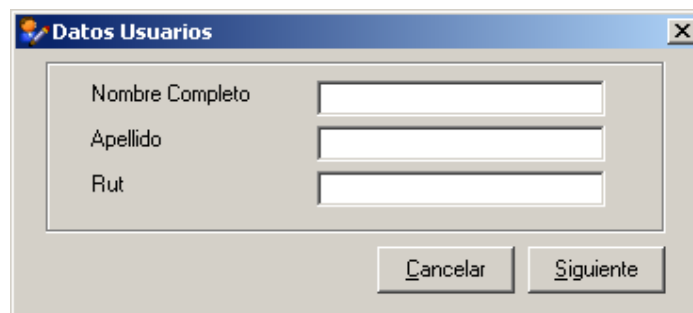


Figura N°29 Diseño de Interfaz de Inserción y Eliminación de Usuarios.

The image shows a Windows-style dialog box titled "Asignación Clave". It contains the following elements:

- Three text input fields labeled "Usuario", "Clave", and "Repita la Clave".
- A checkbox labeled "Administrador" which is currently checked.
- A horizontal separator line.
- A text label below the line: "Importante : Datos de ingreso al sistema del usuario".
- Three buttons at the bottom: "Volver", "Limpiar", and "Terminar".

Figura N°30 Diseño de Interfaz de Inserción y Eliminación de Usuarios.

7.1.2 Implementación y Carga de Base de Datos

7.1.2.1 Diseño Físico de las Tablas Principales en la Base de Datos

- ASOCIADOS

```
#####
# Table: ASOCIADOS
#####
CreateTble C=ASOCIADOS N="Asociados"
(
  C=MAC T=VARCHAR(17) P=Yes M=Yes N="MAC",
  C=NOMPROCES T=VARCHAR(255) P=Yes M=Yes N="NomProces"
);

#####
# Index: SE_ASOCIAN_PK
#####
CreateIndx C=SE_ASOCIAN_PK T=ASOCIADOS U=unique K=primary
(
  C=MAC A=ASC,
  C=NOMPROCES A=ASC
);
```

- COMPUTADOR

```
#####
# Table: COMPUTADOR
#####
CreateTble C=COMPUTADOR N="Computador"
(
  C=NOMBREPC T=VARCHAR(255) P=No M=No N="NombrePC",
  C=MAC T=VARCHAR(17) P=Yes M=Yes N="MAC",
  C=IP T=VARCHAR(15) P=No M=No N="IP",
  C=OBS T=VARCHAR(255) P=No M=No N="OBS"
);

#####
# Index: COMPUTADOR_PK
#####
CreateIndx C=COMPUTADOR_PK T=COMPUTADOR U=unique K=primary
(
  C=MAC A=ASC
);
```

- DATOSUSER

```
#####  
# Table: DATOSUSER  
#####  
CreateTble C=DATOSUSER N="DatosUser"  
(  
    C=RUT T=VARCHAR(9) P=Yes M=Yes N="RUT",  
    C=NOMBRES T=VARCHAR(60) P=No M=No N="Nombres",  
    C=APELLIDOS T=VARCHAR(60) P=No M=No N="Apellidos"  
);  
  
#####  
# Index: DATOSUSER_PK  
#####  
CreateIndx C=DATOSUSER_PK T=DATOSUSER U=unique K=primary  
(  
    C=RUT A=ASC  
);
```

- PERFIL

```
#####  
# Table: PERFIL  
#####  
CreateTble C=PERFIL N="Perfil"  
(  
    C="USER" T=VARCHAR(60) P=Yes M=Yes N="USER",  
    C=RUT T=VARCHAR(9) P=No M=Yes N="RUT",  
    C=CODTIPO T=LONG P=No M=Yes N="CodTipo",  
    C=CLAVE T=VARCHAR(60) P=No M=No N="Clave"  
);  
  
#####  
# Index: PERFIL_PK  
#####  
CreateIndx C=PERFIL_PK T=PERFIL U=unique K=primary  
(  
    C="USER" A=ASC  
);
```

- TIPO

```
#####  
# Table: TIPO  
#####  
CreateTble C=TIPO N="Tipo"  
(  
    C=CODTIPO T=LONG P=Yes M=Yes N="CodTipo",  
    C=NOMTIPO T=VARCHAR(60) P=No M=No N="NomTipo"  
);
```



```

#=====
# Index: TIPO_PK
#=====
CreateIndx C=TIPO_PK T=TIPO U=unique K=primary
(
    C=CODTIPO A=ASC
);

```

- TPROCESS

```

#=====
# Table: TPROCESS
#=====
CreateTble C=TPROCESS N="Tprocess"
(
    C=NOMPROCES T=VARCHAR(255) P=Yes M=Yes N="NomProces",
    C=NAMEPROGM T=VARCHAR(255) P=No M=No N="NameProgm",
    C=PATHPROCCES T=VARCHAR(255) P=No M=No N="PathProcess",
    C=TAMANO T=LONG P=No M=No N="Tamaño"
);

```

```

#=====
# Index: TPROCESS_PK
#=====
CreateIndx C=TPROCESS_PK T=TPROCESS U=unique K=primary
(
    C=NOMPROCES A=ASC
);

```

7.1.2.2 *Diseño Físico de las Relaciones entre las Tablas*

Principales en la Base de Datos

```
CreateJoin C=FK_ASOCIADO_SE_ASOCIA_COMPUTAD T=ASOCIADOS  
P=COMPUTADOR D=restrict U=restrict  
(  
    P=MAC F=MAC  
);
```

```
CreateJoin C=FK_ASOCIADO_SE_ASOCIA_TPROCESS T=ASOCIADOS P=TPROCESS  
D=restrict U=restrict  
(  
    P=NOMPROCES F=NOMPROCES  
);
```

```
CreateJoin C=FK_PERFIL_DEFINE_TIPO T=PERFIL P=TIPO D=restrict  
U=restrict  
(  
    P=CODTIPO F=CODTIPO  
);
```

```
CreateJoin C=FK_PERFIL_POSEE_DATOSUSE T=PERFIL P=DATOSUSER  
D=restrict U=restrict  
(  
    P=RUT F=RUT  
);
```

7.1.2.3 *Diseño Físico de los Procedimientos Almacenados de la Base de Datos*

- BDtsUser

```
PARAMETERS [RutUser] Text;  
SELECT *  
FROM DATOSUSER  
WHERE RUT=[RutUser];
```

- BExistUser

```
PARAMETERS [User] Text;  
SELECT PERFIL.USER  
FROM PERFIL  
WHERE PERFIL.USER=[User];
```

- BExitPC

```
PARAMETERS [MACPC] Text, [IPPC] Text;  
SELECT COMPUTADOR.MAC, COMPUTADOR.IP  
FROM COMPUTADOR  
WHERE COMPUTADOR.MAC=[MACPC] AND COMPUTADOR.IP=[IPPC];
```

- CDesplgPCs

```
SELECT COMPUTADOR.NOMBREPC, COMPUTADOR.IP  
FROM COMPUTADOR;
```

- CDtsCtaUsr

```
PARAMETERS [User] Text, [Passwrđ] Text;  
SELECT PERFIL.USER, PERFIL.CLAVE, PERFIL.CODTIPO  
FROM PERFIL  
WHERE (((PERFIL.USER)=[User]) AND ((PERFIL.CLAVE)=[Passwrđ]));
```

- CListarPC

```
SELECT COMPUTADOR.*  
FROM COMPUTADOR;
```

- CPctaPcrocoss

```
SELECT TProces.*  
FROM TProces;
```

- CProcess

```
PARAMETERS [Proces] Text;  
SELECT *  
FROM TProces  
WHERE TProces.NomProces=[Proces];
```

- CUCrgaUsr

```
SELECT DISTINCTROW DATOSUSER.NOMBRES, DATOSUSER.APELLIDOS,  
DATOSUSER.RUT  
FROM DATOSUSER INNER JOIN PERFIL ON DATOSUSER.RUT =  
PERFIL.RUT;
```

- DeleteProcess

```
PARAMETERS [Proces] Text;  
DELETE *  
FROM TProces  
WHERE NomProces = [Proces];
```

- DltePC

```
PARAMETERS [MACPC] Text;  
DELETE COMPUTADOR.MAC  
FROM COMPUTADOR  
WHERE (((COMPUTADOR.MAC)=[MACPC]));
```

- DltePerfil

```
PARAMETERS [User] Text;  
DELETE PERFIL.USER  
FROM PERFIL  
WHERE (((PERFIL.USER)=[User]));
```

- DlteUser

```
PARAMETERS [RutUser] Text;  
DELETE DATOSUSER.RUT  
FROM DATOSUSER  
WHERE (((DATOSUSER.RUT)=[RutUser]));
```

- DtsCtaUser

```
PARAMETERS [RutUser] Text;  
SELECT *  
FROM PERFIL  
WHERE RUT=[RutUser];
```

- IDtosCuenta

```
PARAMETERS [User] Text, [TpoUser] Long, [RutUser] Text, [Clave] Text;  
INSERT INTO PERFIL ( USER, RUT, CODTIPO, CLAVE )  
SELECT [User] AS Expr1, [RutUser] AS Expr2, [TpoUser] AS Expr3, [Clave] AS  
Expr4;
```

- IncrtPC

```
PARAMETERS [NAMEPC] Text, [MACPC] Text, [IPPC] Text, [OBS] Text;  
INSERT INTO COMPUTADOR ( NOMBREPC, MAC, IP, OBS )  
SELECT [NAMEPC] AS Expr1, [MACPC] AS Expr2, [IPPC] AS Expr3, [OBS] AS  
Expr4;
```

- InsrtDtsPrsn

```
PARAMETERS [RutPersn] Text, [Nom] Text, [Apellido] Text;  
INSERT INTO DATOSUSER ( RUT, NOMBRES, APELLIDOS )  
SELECT [RutPersn] AS Expr1, [Nom] AS Expr3, [Apellido] AS Expr4;
```

- IProcesos

```
PARAMETERS [Proces] Text, [NameProgram] Text;  
INSERT INTO TProces ( NomProces, NameProgram )  
SELECT [Proces] AS Expr1, [NameProgram] AS Expr2;
```

- IUser

```
PARAMETERS [RutUser] Text;  
INSERT INTO PERFIL ( RUT )  
SELECT [RutUser] AS Expr1;
```

- UpDtosPerson

```
PARAMETERS [Rut] Text, [Name] Text, [Apellido] Text;  
UPDATE DATOSUSER SET NOMBRES = [Name], APELLIDOS = [Apellido]  
WHERE DATOSUSER.RUT=[Rut];
```

- UPDtsPC

```
PARAMETERS [NamePC] Text, [MACPC] Text, [IPPC] Text, [OBSPC] Text;  
UPDATE COMPUTADOR SET NOMBREPC = [NamePC], IP = [IPPC], OBS =  
[OBSPC]  
WHERE COMPUTADOR.MAC=[MACPC];
```

7.2 Pruebas

7.2.1 Pruebas del Software

La presente etapa se centra en probar el prototipo con datos ficticios y reales, de este modo se pretende comprobar el buen funcionamiento del mismo. Detectar los errores brinda la posibilidad de corregirlos y revisar la elaboración de la codificación verificando que haya realizado de forma correcta.

En un principio las pruebas se realizaron en el equipo en el que se desarrolló el prototipo, de manera local. La segunda fase de pruebas consistió en utilizar la dirección IP del mismo equipo, y actualizar al prototipo cliente desde el servidor, de esta forma se asegura que se estén grabando los datos correctamente de manera remota.

Una vez hecha estas pruebas, se harán los ajustes necesarios que resulten de las pruebas anteriores, y se implantará definitivamente el sistema.

7.2.1.1 Subsistema Control de Usuarios

7.2.1.1.1 Representación de Casos de Prueba

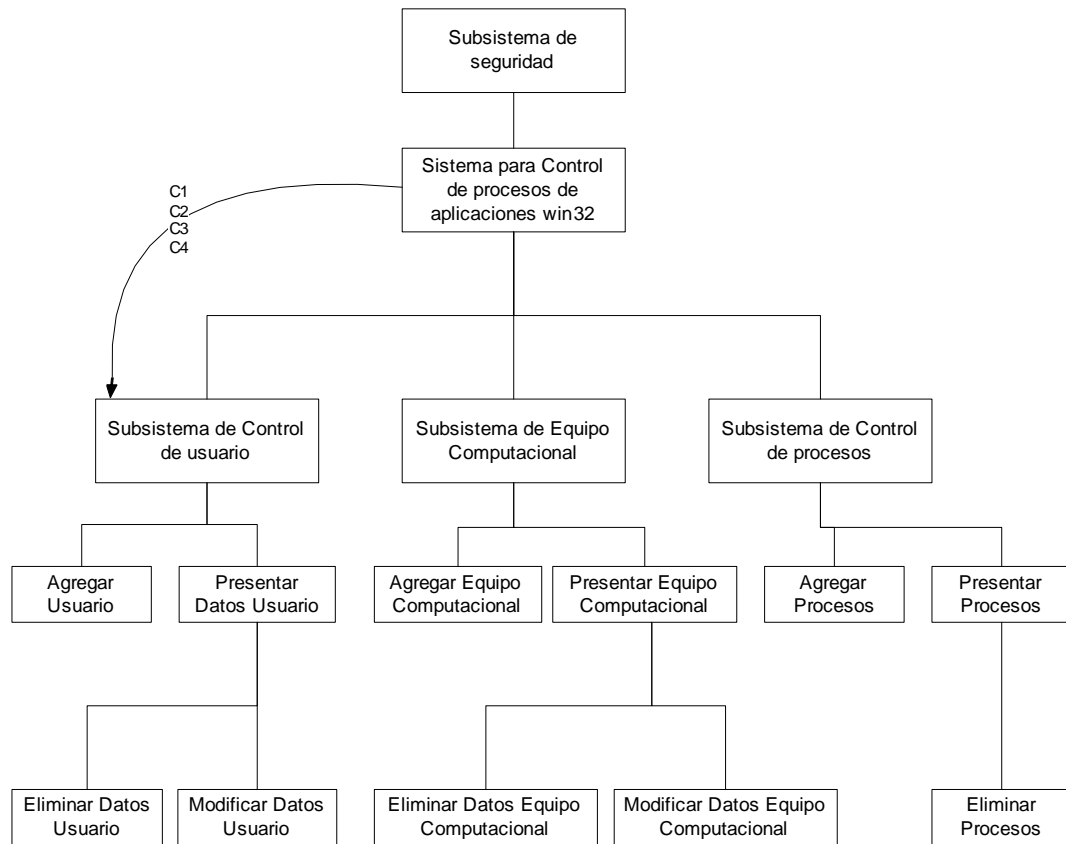


Figura N°31 Diagrama de Caminos Independientes

7.2.1.1.2 Determinación de Caminos Independientes y Casos de Prueba

C1	Ingresar Datos
C2	Consultar Datos
C3	Modificar Datos
C4	Eliminar Datos

- C1
 - Ventana 1
 - Seleccionar menú Archivo
 - Abrir ventana Agregar usuario
 - Ingresar Nombre Completo
 - Ingresar Apellido
 - Ingresar RUT del Usuario
 - No muestra datos
 - Presionar botón Siguiente
 - Ventana 2
 - Ingresar Usuario
 - Ingresar Clave
 - Ingresar Repetir Clave
 - No muestra datos
 - Presionar botón terminar

- C2
 - Seleccionar menú Archivo
 - Abrir ventana Datos Usuario
 - Posicionar cursor del mouse sobre la grilla que muestra el nombre del usuario de interés
 - Hacer clic botón derecho Mouse sobre el nombre del usuario de interés
 - Muestra ventana Datos usuario
 - Seleccionar datos personales o datos cuenta haciendo clic sobre las lengüetas tabuladotas Datos Personales o Datos Cuenta(s)
 - Datos Personales
 - Muestra datos personales usuario
 - Datos Cuenta(s)
 - Muestra datos de la o las cuentas de acceso al sistema que posee el usuario

- C3
 - Realizar pasos C(2)
 - Si se selecciona la lengüeta tabuladota Datos Personales
 - Posicionar cursor sobre botón modificar
 - Hacer clic derecho del Mouse, sobre botón modificar
(Se podrá modificar nombre completo y apellido)
 - Si muestra Datos
 - Si se selecciona Datos Cuenta(s)
 - Posicionar cursor del mouse sobre la grilla que muestra el nombre de la cuenta de usuario de interés
 - Hacer clic botón derecho mouse sobre el nombre de la cuenta de usuario de interés
 - Se despliega un popup
 - Del menú popup seleccionar “Cambiar Contraseña”.
 - No muestra Datos
- C4
 - Realizar pasos C(2)
 - Si se selecciona la lengüeta tabuladota Datos Personales
 - Posicionar cursor sobre botón modificar
 - Hacer clic derecho del Mouse, sobre botón “Eliminar”
 - Despliega mensaje
 - Aceptar
 - No muestra Datos
 - Si se selecciona Datos Cuenta(s)
 - Posicionar cursor del mouse sobre la grilla que muestra el nombre de la cuenta de usuario de interés
 - Hacer clic botón derecho mouse sobre el nombre de la cuenta de usuario de interés
 - Se despliega un popup
 - Del menú popup seleccionar cambiar “Eliminar Usuario”.
 - Despliega mensaje

- Aceptar
- No muestra Datos

7.2.1.1.3 Valores Límite Inserción Datos Usuario

- **Límite Inferior**

Rut	208683810
Name	
Apellido	
LogUser	
Pass	
TpoUser	0

- **Limite Superior**

Rut	20867952K
Name	Variable con 60 caracteres
Apellido	Variable con 60 caracteres
LogUser	Variable con 60 caracteres
Pass	Variable con 60 caracteres
TpoUser	1

- **Valores Intermedios**

Rut	112516735
Name	ABELARDO ADOLFO
Apellido	MALDONADO DOMINGUEZ
LogUser	Jcoronado
Pass	Casitamorada
TpoUser	0

- **Valores Externos a los Límites**

Rut	1721091563
Name	12467
Apellido	4355353
LogUser	8965343
Pass	865568
TpoUser	4

- **Valores Existentes**

Rut	134051744
Name	ARIS DAVID
Apellido	ITURRA SOTO
LogUser	AITURRA
Pass	A
TpoUser	1

7.2.1.1.4 Consultas

Valores Límite Inferior

- Ventana 1
- Entrada “208683810”, “”, “”
- Salida : mensaje de información(“no ha ingresado información, en el campo nombre, apellido o esta es incorrecta”)

Valores Límite Superior

- Ventana 1
 - Entrada
 - “20867952K”
 - “Variable con 60 caracteres”
 - “Variable con 60 caracteres”
 - Salida : Ventana 2
- Ventana 2
 - Entrada
 - “20867952K”,
 - “Variable con 60 caracteres”
 - “Variable con 60 caracteres”
 - “1”
 - Salida : mensaje de información(“ El usuario y su perfil han sido agregados exitosamente”)

Valores Límite Intermedio

- Ventana 1
 - Entrada
 - “112516735”,
 - “ABELARDO ADOLFO”,
 - “MALDONADO DOMINGUEZ”
 - Salida : Ventana 2
- Ventana 2
 - Entrada
 - “112516735”,
 - “Jcoronado”,
 - “Casitamorada”
 - “0”

- Salida : mensaje de información(“ El usuario y su perfil han sido agregados exitosamente”)

Valores Límite Externo a los Limites

- Ventana 1
- Entrada
 - “1721091563”,
 - “12467”,
 - “4355353”
- Salida : mensaje de información(“Los valores ingresados no corresponden a los tipos de datos asociados con los campos”)

Valores Existentes

- Ventana 1
- Entrada
 - “134051744”,
 - “ARIS DAVID”,
 - “ITURRA SOTO”
- Salida : mensaje de información(“El usuario ya existe”)

7.2.1.1.5 Valores Límite Consultar Datos Usuario

- **Límite inferior**

Rut 208683810

- **Limite superior**

Rut 20867952K

- **Valores intermedios**

Rut 112516735

- **Valores No Existentes**

Rut 134051744

7.2.1.1.6 Consultas

Valores Límite Inferior

Lengüeta tabuladota "Datos Personales"

- Entrada

 "208683810"

- Salida : mensaje de información(" No existen datos para este usuario, o fue mal escrito").

Valores Límite Superior

Lengüeta tabuladota "Datos Personales"

- Entrada
"20867952K",
- Salida :

Datos Personales

- Variable con 60 caracteres
- Variable con 60 caracteres

Datos Cuenta(s)

- Variable con 60 caracteres
- Variable con 60 caracteres
- 1

Valores Límite Intermedio

Lengüeta tabuladota "Datos Personales"

- Entrada
"112516735",
- Salida :

Datos Personales

- ABELARDO ADOLFO
- MALDONADO DOMINGUEZ

Datos Cuenta(s)

- Jcoronado

- Casitamorada
- 0

Valores No Existentes

Lengüeta tabuladota “Datos Personales”

- Entrada
“187528763”
- Salida :

Datos Personales

- ABELARDO ADOLFO
- MALDONADO DOMÍNGUEZ

Datos Cuenta(s)

- Jcoronado
- Casitamorada
- 0

7.2.1.1.7 Valores Límite Modificar Datos Usuario

- **Límite Inferior**

Rut 208683810

Name

Apellid

LogUser

Pass-old

Pass

TpoUser 0

- **Limite Superior**

Rut	20867952K
Name	Variable con 60 caracteres
Apellid	Variable con 60 caracteres
LogUser	Variable con 60 caracteres
Pass-old	Variable con 60 caracteres
Pass	Variable con 60 caracteres
TpoUser	1

- **Valores Intermedios**

Rut	112516735
Name	ABELARDO ADOLFO
Apellid	MALDONADO DOMINGUEZ
LogUser	Jcoronado
Pass-old	Casitamorada
Pass	Casita
TpoUser	0

7.2.1.1.8 Consultas

Valores Límite Inferior

Lengüeta tabuladota “Datos Personales”

- Entrada
 “208683810”,
 “”
 “”
- Salida : mensaje de información(“no ha ingresado información, en el campo nombre, apellido o esta es incorrecta”)

Valores Límite Superior

Lengüeta tabuladota “Datos Personales”

- Entrada
 “20867952K”
 “Variable con 60 caracteres”
 “Variable con 60 caracteres”
- Salida : mensaje de información(“ los datos del usuario han sido modificados exitosamente”)

Lengüeta tabuladota “Datos Cuenta(s)”

- Entrada
 “Variable con 60 caracteres”
 “Variable con 60 caracteres”
 “Variable con 60 caracteres”
 “1”

- Salida : mensaje de información(“ La clave ha sido cambiada exitosamente”)

Valores Límite Intermedio

Lengüeta tabuladota “Datos Personales”

- Entrada
“112516735”,
“ABELARDO ADOLFO II”,
“MALDONADO DOMINGUEZ”
- Salida : mensaje de información(“ los datos del usuario han sido modificados exitosamente”)

Lengüeta tabuladota “Datos Cuenta(s)”

- Entrada
“Jcoronado”,
“Casitamorada”
“Casita”
“0”
- Salida : mensaje de información(“ La clave ha sido cambiada exitosamente”)

7.2.1.1.9 Valores Límite Eliminar Datos Usuario

- **Límite Inferior**

Rut 208683810

LogUser

- **Limite Superior**

Rut 20867952K

LogUser Variable con 60 caracteres

- **Valores Intermedios**

Rut 112516735

LogUser Jcoronado

- **Valores No Existentes**

Rut 134051744

LogUser AITURRA

7.2.1.1.10 Consultas

Valores Límite Inferior

Lengüeta tabuladota “Datos Personales”

- Entrada
“208683810”
- Salida : mensaje de información (“El Usuario ha sido eliminada exitosamente”).

Lengüeta tabuladota “Datos Cuenta(s)”

- Entrada
“”
- Salida : mensaje de información (“ Campo vacío, o no existe usuario que eliminar ”)

Valores Límite Superior

Lengüeta tabuladota “Datos Personales”

- Entrada
“20867952K”
- Salida : mensaje de información (“El Usuario ha sido eliminada exitosamente”).

Lengüeta tabuladota “Datos Cuenta(s)”

- Entrada
“Variable con 60 caracteres”

- Salida : mensaje de información (“El perfil de Usuario ha sido eliminada exitosamente”).

Valores Límite Intermedio

Lengüeta tabuladota “Datos Personales”

- Entrada
“112516735”
- Salida : mensaje de información (“El Usuario ha sido eliminada exitosamente”).

Lengüeta tabuladota “Datos Cuenta(s)”

- Entrada
“Jcoronado”
- Salida : mensaje de información (“El perfil de Usuario ha sido eliminada exitosamente”).

Valores No Existentes

Lengüeta tabuladota “Datos Personales”

- Entrada
“187528763”
- Salida : mensaje de información (“Este Usuario no existe, o ha sido mal escrito”).

Lengüeta tabuladota “Datos Cuenta(s)”

- Entrada
“acoronado”
- Salida : mensaje de información (“Este Perfil de Usuario no existe, o ha sido mal escrito”).

7.2.1.2 Subsistema Computador Personal

7.2.1.2.1 Representación de Casos de Prueba

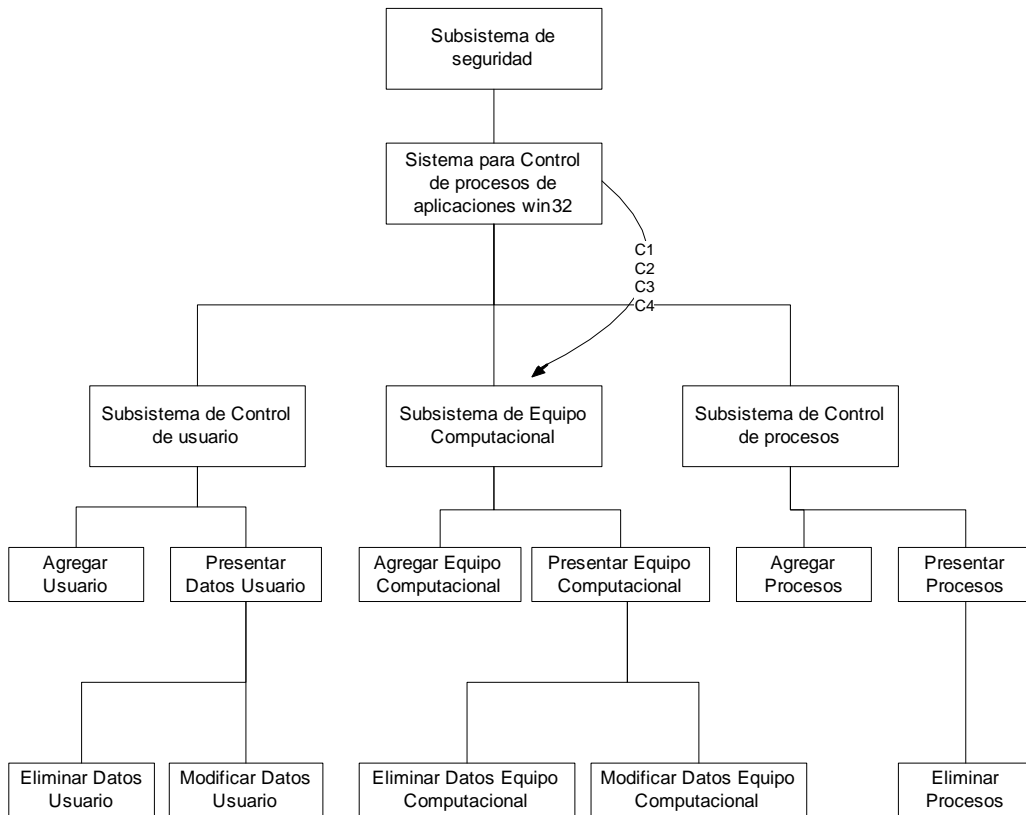


Figura N°32 Diagrama de Caminos Independientes

7.2.1.2.2 Determinación de Caminos Independientes y Casos de Prueba

C1	Ingresar Datos
C2	Consultar Datos
C3	Modificar Datos
C4	Eliminar Datos

- C1
 - Seleccionar menú Archivo
 - Abrir ventana Agregar PC
 - Ingresar Nombre PC
 - MAC
 - IP
 - OBS
- C2
 - Seleccionar menú Archivo
 - Abrir ventana Datos PC
 - Muestra ventana Datos PC
- C3
 - Realizar pasos C(2)
 - Posicionar cursor del mouse sobre la grilla que muestra el nombre del PC de interés
 - Hacer clic botón derecho del mouse sobre el nombre del PC de interés
 - Se despliega un popup
 - Del menú popup seleccionar "Modificar".
 - muestra Datos
- C4
 - Realizar pasos C(2)
 - Posicionar cursor del mouse sobre la grilla que muestra el nombre del PC de interés
 - Hacer clic botón derecho del mouse sobre el nombre del PC de interés
 - Se despliega un popup
 - Del menú popup seleccionar "Eliminar".
 - No muestra Datos

7.2.1.2.3 Valores Límite Inserción Computador Personal

- **Límite Inferior**

MAC	00-00-00-00-00-00
NamePC	A
IPPC	1.1.1.1
OBSPC	

- **Limite Superior**

MAC	FF-FF-FF-FF-FF-FF
NamePC	Variable con 255 caracteres
IPPC	255.255.255.255
OBSPC	Variable con 255 caracteres

- **Valores Intermedios**

MAC	00-07-95-EC-2E-7E
NamePC	PC1
IPPC	192.168.1.34
OBSPC	Pesa1

- **Valores Externos a los Límites**

MAC	00-01-4A-F2-46-ZZ
NamePC	""
IPPC	10.0.0.1.6
OBSPC	Variable con 300 caracteres

- **Valores Existentes**

MAC	00-01-4A-F2-46-69
NamePC	PC0
IPPC	10.0.0.1
OBSPC	Server

7.2.1.2.4 Consultas

Valores Límite Inferior

- Entrada

“00-00-00-00-00-00”

“a”

“1.1.1.1”

“”

- Salida : mensaje de información (“La información ha sido ingresada exitosamente”)

Valores Límite Superior

- Entrada

“FF-FF-FF-FF-FF-FF”

“Variable con 255 caracteres”

“255.255.255.255”

“Variable con 255 caracteres”

- Salida : mensaje de información (“La información ha sido ingresada exitosamente”)

Valores Límite Intermedio

- Entrada
 - “00-07-95-EC-2E-7E”
 - “PC1”
 - “192.168.1.34”
 - “Pesa1”
- Salida : mensaje de información(“La información ha sido ingresada exitosamente”)

Valores Límite Externo a los Limites

- Entrada
 - “00-01-4A-F2-46-ZZ”,
 - “”
 - “10.0.0.1.6”
 - “Variable con 300 caracteres”
- Salida : mensaje de información(“Alguno de los campos contienen valores que no corresponden a los tipos de datos asociados con estos”)

Valores Existentes

Entrada

- “00-01-4A-F2-46-69”,
 - “”
 - “10.0.0.1.6”
 - “Variable con 255 caracteres”
- Salida : mensaje de información(“Este PC ya se encuentra en la base de datos”)

7.2.1.2.5 Valores Límite Modificar Datos Usuario

- **Límite Inferior**

MAC	00-00-00-00-00-00
NamePC	Aa
IPPC	1.1.1.1
OBSPC	A

- **Límite Superior**

MAC	FF-FF-FF-FF-FF-FF
NamePC	Carácter + Variable con 254 caracteres
IPPC	255.255.255.255
OBSPC	Carácter + Variable con 254 caracteres

- **Valores Intermedios**

MAC	00-07-95-EC-2E-7E
NamePC	PC2
IPPC	192.168.1.34
OBSPC	Pesa4

- **Valores Externos a los Límites**

MAC	00-01-4A-F2-46-ZZ
NamePC	""
IPPC	10.0.0.1.6
OBSPC	Variable con 300 caracteres

- **Valores Existentes**

MAC	00-01-4A-F2-46-69
NamePC	PC0
IPPC	10.0.0.1
OBSPC	Server

7.2.1.2.6 Consultas

Valores Límite Inferior

- Entrada
 - “00-00-00-00-00-00”
 - “aa”
 - “1.1.1.1”
 - “a”
- Salida : mensaje de información (“La información ha sido Actualizada exitosamente”)

Valores Límite Superior

- Entrada
 - “FF-FF-FF-FF-FF-FF”
 - “Carácter + Variable con 254 caracteres”
 - “255.255.255.255”
 - “Carácter + Variable con 254 caracteres”
- Salida : mensaje de información (“La información ha sido Actualizada exitosamente”)

Valores Límite Intermedio

- Entrada
 - “00-07-95-EC-2E-7E”
 - “PC2”
 - “192.168.1.34”
 - “Pesa4”
- Salida : mensaje de información(“La información ha sido actualizada exitosamente”)

Valores Límite Externo a los Limites

- Entrada
 - “00-01-4A-F2-46-ZZ”,
 - “”
 - “10.0.0.1.6”
 - “Variable con 300 caracteres”
- Salida : mensaje de información(“Alguno de los campos contienen valores que no corresponden a los tipos de datos asociados con estos”)

Valores Límite Existentes

Entrada

- “00-01-4A-F2-46-69”,
 - “PC0”
 - “10.0.0.1”
 - “Serve”
- Salida : mensaje de información(“Estos datos ya se encuentra en la base de datos”)

7.2.1.2.7 Valores Límite Eliminar Datos Usuario

- **Límite Inferior**

MAC	00-00-00-00-00-00
IPPC	1.1.1.1

- **Límite Superior**

MAC	FF-FF-FF-FF-FF-FF
IPPC	255.255.255.255

- **Valores Intermedios**

MAC	00-07-95-EC-2E-7E
IPPC	192.168.1.34

- **Valores Externos a los Límites**

MAC	00-01-4A-F2-46-ZZ
IPPC	10.0.0.1.6

- **Valores No Existentes**

MAC	00-01-4A-F2-46-69
IPPC	10.0.0.1

7.2.1.2.8 Consultas

Valores Límite Inferior

- Entrada
 “00-00-00-00-00-00”
 “1.1.1.1”
- Salida : mensaje de información (“La información ha sido actualizada exitosamente”)

Valores Límite Superior

- Entrada
 “FF-FF-FF-FF-FF-FF”
 “255.255.255.255”
- Salida : mensaje de información (“La información ha sido actualizada exitosamente”)

Valores Límite Intermedio

- Entrada
 “00-07-95-EC-2E-7E”
 “192.168.1.34”
- Salida : mensaje de información (“La información ha sido actualizada exitosamente”)

Valores Límite Externo a los Limites

- Entrada
 “00-01-4A-F2-46-ZZ”,
 “10.0.0.1.6”

- Salida : mensaje de información("Alguno de los campos contienen valores que no corresponden a los tipos de datos asociados con estos o han sido mal escritos")

Valores No Existentes

- Entrada
 - "00-01-4A-F2-46-69"
 - "10.0.0.1.6"
- Salida : mensaje de información("Alguno de los campos contienen valores que no corresponden a los tipos de datos asociados con estos")

7.2.1.3 Subsistema de Control de Procesos

7.2.1.3.1 Representación de Casos de Prueba

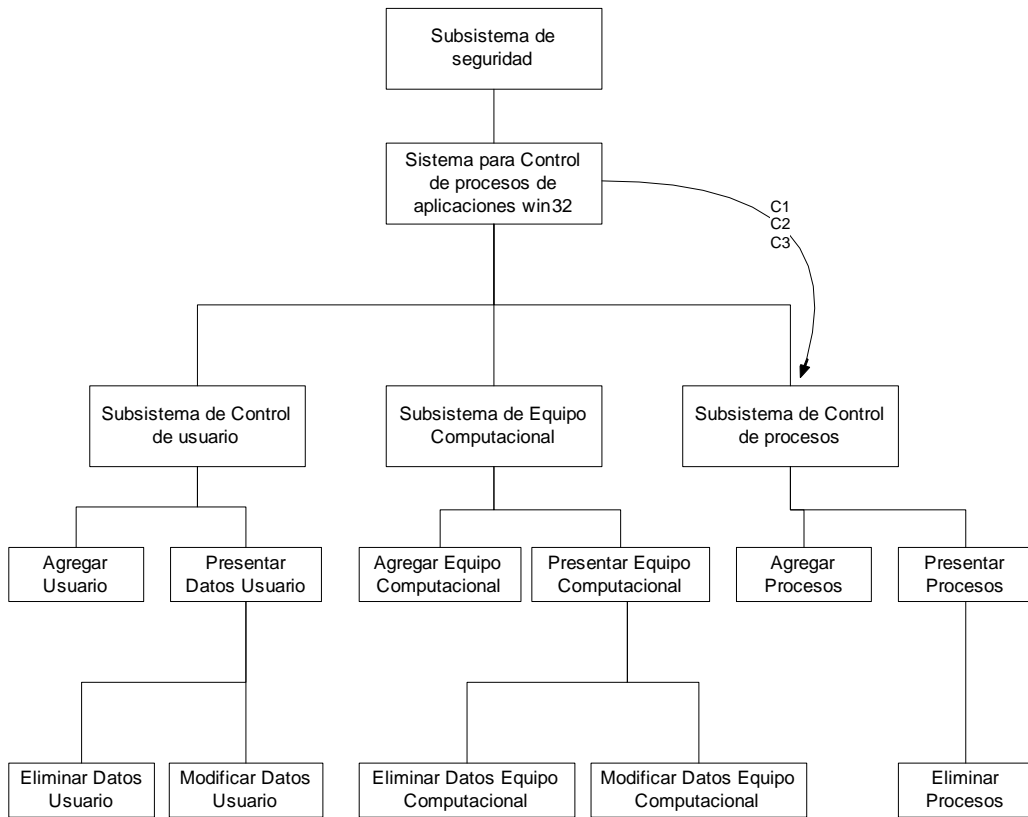


Figura N°33 Diagrama de Caminos Independientes

7.2.1.3.2 Determinación de Caminos Independientes y Casos de Prueba

C1	Ingresar Datos
C2	Consultar Datos
C3	Eliminar Datos

- C1
 - Seleccionar el icono correspondiente al PC que desea administrar (Por defecto presenta los procesos del PC en el que se ejecuta la aplicación)sz4Seleccionar desde la ventana de procesos nuevos aquel que corresponde a la aplicación que se desea ejecutar.
 - Mensaje: ingrese nombre del software que desea que se ejecute.
 - Ingresar Nombre de Aplicación que se desea ejecutar.
- C2
 - Seleccionar el icono correspondiente al PC que desea administrar (Por defecto presenta los procesos del PC en el que se ejecuta la aplicación de control)
- C3
 - Realizar pasos C2
 - Posicionar cursor del mouse sobre la grilla que muestra el nombre del proceso de interés
 - Hacer clic botón derecho del mouse sobre el nombre del nombre del proceso de interés
 - Se despliega un popup
 - Del menú popup seleccionar “eliminar”.
 - No muestra Datos

7.2.1.3.3 Valores Límite Inserción Subsistema de Control de Procesos

- **Límite Inferior**

NameProgrm	A
NameProces	A
Path	A
Tamano	1

- **Límite Superior**

NameProgrm Variable con 255 caracteres
NameProces Variable con 255 caracteres
Path "C:\Archivos de programa\m\fl\.....\a.exe" (255 caracteres)
Tamano 2147483647

- **Valores Intermedios**

NameProgrm Calculadora
NameProces calc.exe
Path "C:\WINDOWS\system32\calc.exe"
Tamano 115.200

- **Valores Externos a los Límites**

NameProgrm Variable con 256 caracteres
NameProces Variable con 300 caracteres
Path "C:\Archivos de programa\m\fl\.....\a.exe" (260 caracteres)
Tamano 2.147.483.649

- **Valores Existentes**

NameProgrm Exel de Office
NameProces EXCEL.EXE
Path C:\Archivos de programa\Microsoft Office\OFFICE11\
EXCEL.EXE
Tamano 10.199.040

7.2.1.3.4 Consultas

Valores Límite Inferior

- Entrada
 - “a”
 - “a”
 - “a”
 - “1”
- Salida : mensaje de información (“La información ha sido ingresada exitosamente”)

Valores Límite Superior

- Entrada
 - “Variable con 255 caracteres”
 - “Variable con 255 caracteres
 - “C:\Archivos de programa\mf\.....\a.exe” (255 caracteres)
 - “Variable con 255 caracteres”
 - “2147483647”
- Salida : mensaje de información (“La información ha sido ingresada exitosamente”)

Valores Límite Intermedio

- Entrada
 - “Calculadora”
 - “calc.exe”
 - “C:\WINDOWS\system32\calc.exe”
 - “115.200”

- Salida : mensaje de información("La información ha sido ingresada exitosamente")

Valores Límite Externo a los Limites

- Entrada
 - "Variable con 256 caracteres"
 - "Variable con 300 caracteres"
 - "C:\Archivos de programa\m\fl\.....\a.exe" (260 caracteres)"
 - "2.147.483.649"
- Salida : mensaje de información("Alguno de los campos contienen valores que no corresponden a los tipos de datos asociados con estos")

Valores Existentes

Entrada

- "Exel de Office"
 - "EXCEL.EXE"
 - "C:\Archivos de programa\Microsoft Office\OFFICE11\ EXCEL.EXE"
 - "10.199.040"
- Salida : mensaje de información("ya se encuentra en la base de datos esta información")

7.2.1.3.5 Valores Límite Consultar Subsistema de Control de Procesos

- **Límite Inferior**

MAC 00-00-00-00-00-00

- **Límite Superior**

MAC FF-FF-FF-FF-FF-FF

- **Valores Intermedios**

MAC 00-07-95-EC-2E-7E

- **Valores Externos a los Límites**

MAC 00-01-4A-F2-46-ZZ

- **Valores No Existentes**

MAC 00-01-4A-F2-46-69

7.2.1.3.6 Consultas

Valores Límite Inferior

- Entrada
“00-00-00-00-00-00”
- Salida : Lista de procesos validos para este Computador

Valores Límite Superior

- Entrada
“FF-FF-FF-FF-FF-FF”
- Salida : Lista de procesos validos para este Computador

Valores Límite Intermedio

- Entrada
“00-07-95-EC-2E-7E”
- Salida : Lista de procesos validos para este Computador

Valores Límite Externo a los Limites

- Entrada
“00-01-4A-F2-46-ZZ”,
- Salida : mensaje de información(“El valor del dato ingresado, no corresponden a los tipos de datos asociados con este, o bien han sido mal escritos”)

Valores Existentes

- Entrada
"00-01-4A-F2-46-69"
- Salida : mensaje de información("Ya existen datos asociados a este valor en la base de datos")

7.2.1.3.7 Valores Límite Eliminar Subsistema de Control de Procesos

- **Límite Inferior**

NameProgrm	A
NameProces	A
Path	A
Tamano	1

- **Límite Superior**

NameProgrm	Variable con 255 caracteres
NameProces	Variable con 255 caracteres
Path	"C:\Archivos de programa\m\fl\.....\a.exe" (255 caracteres)
Tamano	2147483647

- **Valores Intermedios**

NameProgrm	Calculadora
NameProces	calc.exe
Path	"C:\WINDOWS\system32\calc.exe"
Tamano	115.200

- **Valores Externos a los Límites**

NameProgrm	Variable con 256 caracteres
NameProces	Variable con 300 caracteres
Path	“C:\Archivos de programa\m\fl\.....\a.exe” (260 caracteres)
Tamano	2.147.483.649

- **Valores Existentes**

NameProgrm	Exel de Office
NameProces	EXCEL.EXE
Path	C:\Archivos de programa\Microsoft Office\OFFICE11\ EXCEL.EXE
Tamano	10.199.040

7.2.1.3.8 Consultas

Valores Límite Inferior

- Entrada
 - “a”
 - “a”
 - “a”
 - “1”
- Salida : mensaje de información (“La información ha sido eliminada exitosamente”)

Valores Límite Superior

- Entrada
 - “Variable con 255 caracteres”
 - “Variable con 255 caracteres)”
 - “C:\Archivos de programa\m\fl\.....\a.exe” (255 caracteres)
 - “Variable con 255 caracteres”
 - “2147483647”
- Salida : mensaje de información(“La información ha sido eliminada exitosamente”)

Valores Límite Intermedio

- Entrada
 - “Calculadora”
 - “calc.exe”
 - “C:\WINDOWS\system32\calc.exe”
 - “115.200”
- Salida : mensaje de información(“La información ha sido eliminada exitosamente”)

Valores Límite Externo a los Limites

- Entrada
 - “Variable con 256 caracteres)”
 - “Variable con 300 caracteres
 - “C:\Archivos de programa\m\fl\.....\a.exe” (260 caracteres)”
 - “2.147.483.649”
- Salida : mensaje de información(“Alguno de los campos contienen valores que no corresponden a los tipos de datos asociados con estos”)

Valores No Existentes

Entrada

“Excel de Office”

“EXCEL.EXE”

“C:\Archivos de programa\Microsoft Office\OFFICE11\ EXCEL.EXE”

“10.199.0.40”

- Salida : mensaje de información(“ La información requerida no se encuentra en la base de datos, o esta mal escrita”)

7.3 Análisis del Estado Operacional II de los PCs

Tabla N°13 Tiempo que toma la reparación, medido en hora laborables, prototipo ya instalado

Equipo	Mes 7	Mes 8	Mes 9	Mes 10	Mes 11	Mes 12
Pc1	1	0	0	0	0	0,1
Pc2	0	0	0,7	0	0	0
Pc3	0	0,5	0	0,15	0	0
Pc4	0,2	0	0	0	0	0

Tabla N°14 Tiempo existente entre las mantenciones, medido en días laborables, prototipo ya instalado

PC1	Fechas	D Mantenciones
Mes7	04/07/2005	17
Mes12	22/12/2005	121
Promedio		69

PC2	Fechas	D Mantenciones
Mes9	21/09/2005	90
Promedio		90

PC3	Fechas	D Mantenciones
Mes8	24/08/2005	46
Mes11	10/11/2005	55
Promedio		51

PC4	Fechas	D Mantenciones
Mes7	05/07/2005	29
Promedio		29

Tabla N°15 Costos por concepto de mantenciones, Con Aplicación

PC1	Fechas	Costo	Observación
Mes7	04/07/2005	12.000	Mantención Software Pesas
Mes 12	22/12/2005	7.000	Actualización Antivirus
Total		19.000	

PC2	Fechas	Costo	Observación
Mes 11	21/09/2005	7.000	Actualizacion Antivirus
Total		7.000	

PC3	Fechas	Costo	Observación
Mes8	24/08/2005	7.000	Actualización Antivirus
Mes9	10/11/2005	0	Mantención Preventiva Software Contable
Total		7.000	

PC4	Fechas	Costo	Observación
Mes7	05/07/2005	12.000	Mantención Software Pesas
Total		12.000	

8. EVALUACIÓN Y COMPARACIÓN DE LAS PRUEBAS OBTENIDAS DE LOS ESTADOS OPERACIONALES (I Y II) DE LOS PCS.

8.1 Evaluación del Estado Operacional de los PCs

Tabla N°16 Sumatoria de los tiempos, en días laborales, que tarda la mantención

Equipo	Sum 2° Semtre
Pc1	1,1
Pc2	0,7
Pc3	0,65
Pc4	0,2

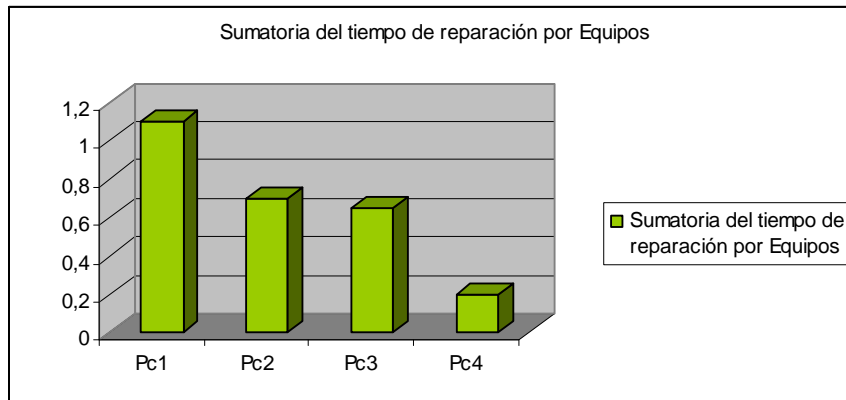


Gráfico N°6 Sumatoria de los Tiempos en Días Laborales, Periodo que Tarda la Mantención

Tabla N°17 Sumatoria del tiempo, en días laborales, que tarda la reparación de los equipos, vista por meses

	Mes 7	Mes 8	Mes 9	Mes 10	Mes 11	Mes 12
PC-Sum-2	1,2	0,5	0,7	0,15	0	0,1

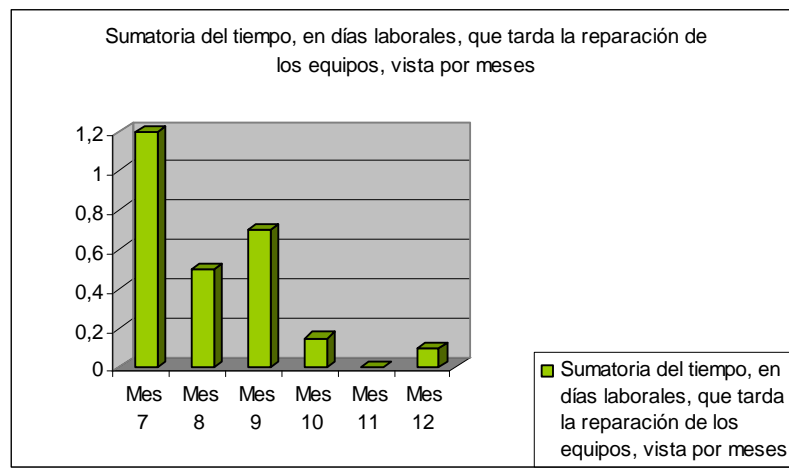


Gráfico N°7 Sumatoria del Tiempo en Días Laborales que Tarda la Reparación de los Equipos, vista por Meses

Tabla N°18 Promedio de días existentes entre mantenciones, vista por PCs.

Segundo Semestre	
PC1	69
PC2	90
PC3	51
PC4	29

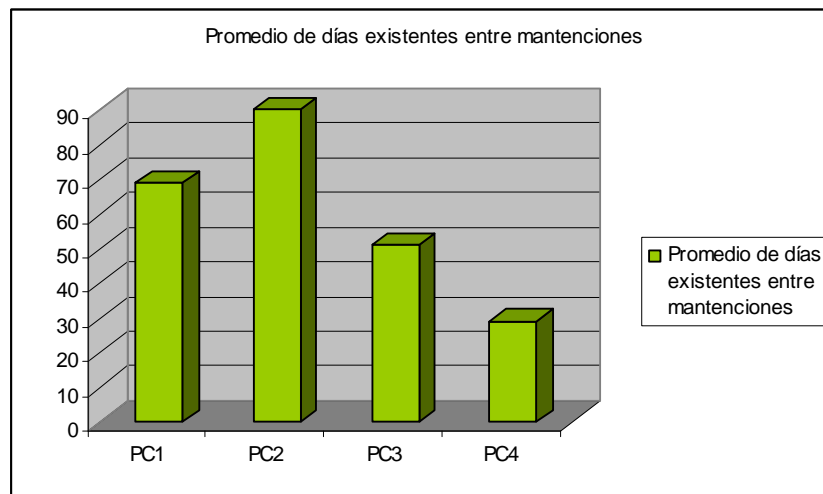


Gráfico N°8 Promedio de Días Existentes entre Mantenciones, Vista por PCs

Tabla N°19 Sumatoria Costos por mantención según PC

Segundo Semestre	
PC1	19.000
PC2	7.000
PC3	7.000
PC4	12.000

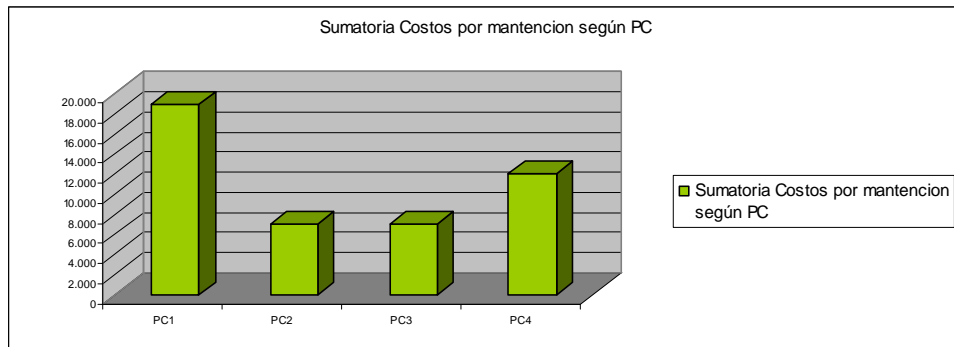


Gráfico N°9 Sumatoria Costos por Mantención según PC

Tabla N°20 Sumatoria Costos por mantención según Mes

	Mes 7	Mes 8	Mes 9	Mes 10	Mes 11	Mes 12
Total por meses	24.000	7.000	0	0	7.000	7.000

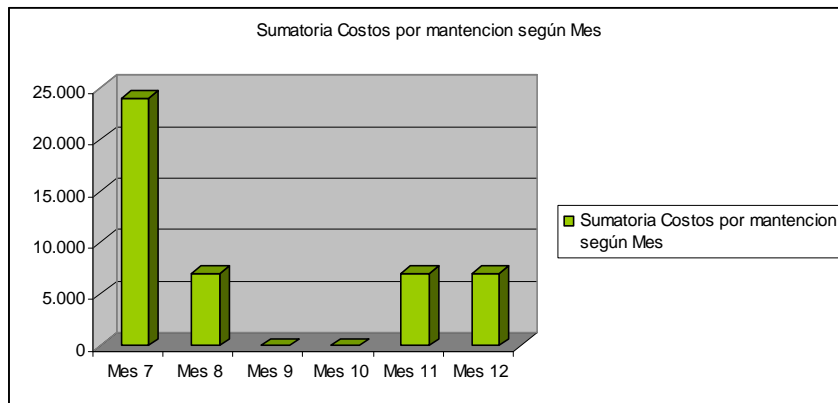


Gráfico N°10 Sumatoria Costos por Mantención según Mes

8.2 Comparación de las Pruebas Obtenidas de los Estados Operacionales (I y II) de los PCs

Tabla N°21 Sumatoria Total del Tiempo en días laborales, que toman las Reparaciones

Primer Semestre	155
Segundo Semestre	2,65

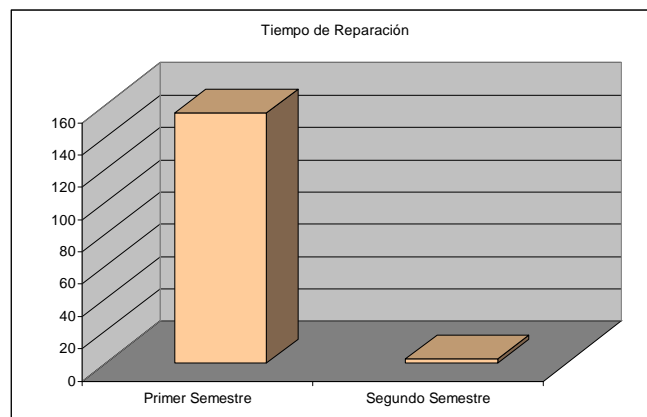


Gráfico N°11 Sumatoria Total del Tiempo en Días Laborales, que Toman las Reparaciones

Tabla N°22 Tiempo existente entre las mantenciones, Promedio semestral, medido en días laborales

Primer Semestre	25
Segundo Semestre	60

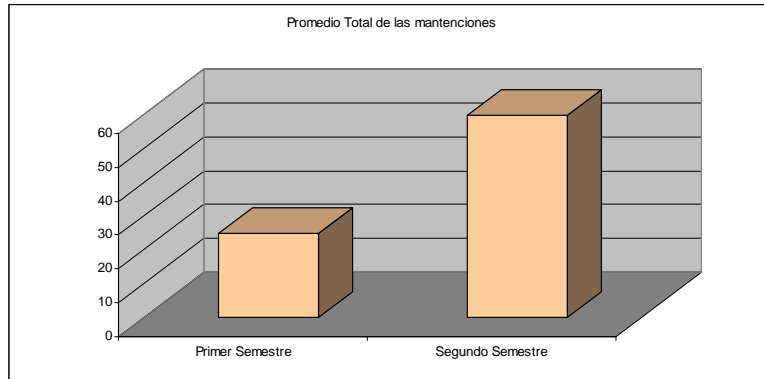


Gráfico N°12 Tiempo Existente entre las Mantenciones, Promedio Semestral, Medido en Días Laborales

Tabla N°23 Sumatoria Total de los costos de Mantención

Primer Semestre	233.000
Segundo Semestre	45.000

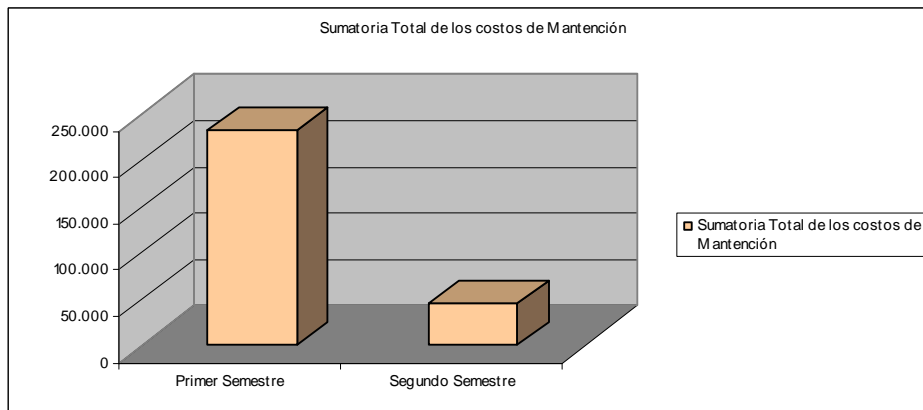


Gráfico N°13 Sumatoria Total de los costos de Mantención

9. CONCLUSIONES Y/O RECOMENDACIONES

En este seminario se desarrolló una solución factible, viable y sencilla de operar, para evitar ciertos fallos computacionales asociados a software, tales como corrupción de programas, la implantación de software malicioso etc.

Orientar la solución de fallos asociados a software a la causa y no al problema generado por ella. Representa un enfoque diferente y práctico para su implementación

La utilización de esta solución a través de una aplicación o prototipo de sistema computacional, no sólo permitió mejorar el desempeño general de los computadores y reducir las molestias asociadas a los inconvenientes surgidos del funcionamiento inadecuado de los software, si no también reducir los tiempos de servicio técnico de la máquina.

Una etapa fundamental es la identificación de procesos válidos que nutren el sistema, para ello bastará con identificar los procesos de sistema, los de operación y los de gestión; donde los primeros son generados por el sistema operativo, los segundos por los software de operación (ej: Excel, photoshop, lotus, etc.) y los últimos por los programas inherentes al área de producción de la empresa o institución.

Cabe destacar que este seminario de titulación ha permitido poner en práctica los conocimientos adquiridos como alumno, en una situación

concreta surgida en la etapa laboral. Diseñar y construir un prototipo, independiente del motivo por el cual debe ser implementado, no resulta simple pues requiere de un buen estudio previo, revisando detalles y situaciones diversas; sin embargo, apoyándose en las herramientas, metodologías y modelos disponibles para este fin fue posible la construcción del prototipo funcional.

Finalmente, las mejoras que consolidarán el prototipo en una aplicación sólida, están determinadas por el mejoramiento del código, depuración de ventanas, incorporación de ayudas y otras funcionalidades tales como la captura de los datos del usuario que opera el computador. Con ello existirá un replanteamiento del modelo E-R pues se pretende establecer una relación entre Usuario y Computador personal.

10. BIBLIOGRAFÍA

[monografías2005] Sin Autor. Procesos y Subprocesos en Windows.

Disponible en:

<http://www.monografias.com/trabajos14/administ-procesos/administ-procesos.shtml#PROCE>.

[rincondelvago] Sin Autor. Metodología de investigación. Disponible en:

http://html.rincondelvago.com/metodologia-de-investigacion_1.html

[MSDN Library] Slasher Keeper. Procesos y Subprocesos en Windows.

Disponible en:

<http://foro.elhacker.net/index.php/topic,62754.msg287401.html#msg287401>, 21 Marzo 2005, 07:18.

[Pressman1993] Pressmann, Roger. Ingeniería del Software: Un enfoque práctico. McGraw-Hill. Tercera edición. 1993

[Métodos "I + D" de la Informática] Graciela Elisa Barchini. Métodos "I + D" de la Informática. 2 de agosto 2005

[Grupo KYBELE] Esperanza Marcos. Investigación en Ingeniería del Software vs.Desarrollo Software.

11. ANEXOS

11.1 ¿Qué es un Proceso?

Un proceso es una entidad dinámica que nace cuando se carga un programa en memoria y muere al finalizar la ejecución del programa cargado en memoria.

Un proceso está compuesto por el código ejecutable, una sección de datos que contienen las variables globales, una sección de stack o pila que contiene datos temporales (tales como: parámetros de subrutinas, direcciones de retorno y variables temporales) y estado de los registros del procesador. El programa corresponde a una entidad pasiva, en cambio el proceso corresponde a una entidad activa.

Hablando de los recursos que utilizan ambos conceptos podemos decir, que el programa utiliza únicamente memoria secundaria (disco duro, CD, DVD, Diskette), en cambio el proceso utiliza memoria principal (RAM) y procesador. En la siguiente figura se aprecia ambos conceptos.

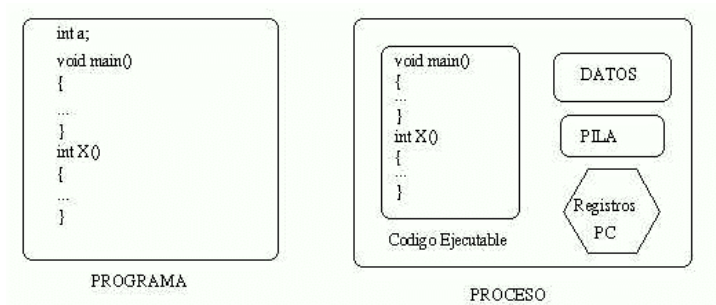


Figura. N°34 Diferencia entre Programa y Proceso, <http://www2.udec.cl/~elozada/capitulo II.htm>

Definir el proceso como una entidad significa que todo proceso tiene un nombre, tiene un identificador, que lo diferencia de todos los otros y que sirve para individualizarlo. Que la entidad sea dinámica significa que el proceso tiene vida: nace, vive y muere.

La vida de un proceso se puede representar con el siguiente diagrama de estados de un proceso:

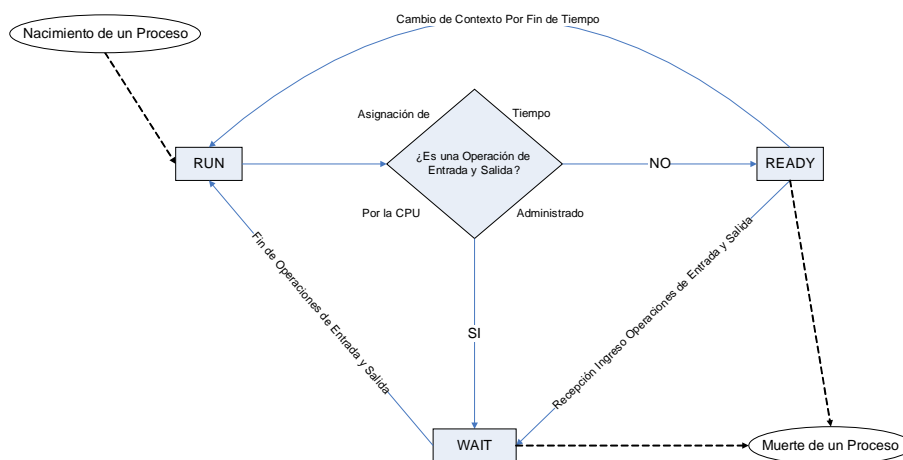


Figura. N°35 Vida de un Proceso, diseño para proyecto

En un sistema monoprocesador sólo uno de todos los procesos que pueden existir estará en RUN (en ejecución), en este estado, la CPU estará ejecutando las instrucciones correspondientes al programa asociado al proceso. Cuando el proceso sale del estado de RUN para pasar al estado de READY (preparado) o de WAIT (en espera), otro proceso que este en READY pasará a RUN, produciéndose un cambio de contexto. Esto significa que el sistema operativo debe salvar toda la información relativa al proceso que estaba en RUN (el sistema se guarda una imagen estática de como estaba), para que cuando, posteriormente, vuelva a estar en RUN pueda reemprender el trabajo en el mismo lugar. Esta operación se repite continuamente.

Para ejecutar todos los procesos que pueden existir, el sistema irá asignando diferentes tiempos de CPU (quantum) a cada uno de los procesos. Cuando a un proceso se le acabe su tiempo, pasará al estado de READY y permanecerá en él hasta que le vuelva a ser asignada la CPU. Cuando un proceso deba realizar una operación de entrada/salida pasará al estado de WAIT y permanecerá en él hasta que dicha operación se acabe, pasando en ese momento al estado de READY.

11.1.1 *Procesos y Subprocesos*

Las aplicaciones basadas en Win32 consisten de uno o más procesos. Un subproceso es la unidad básica para la cual el sistema operativo asigna tiempo del procesador. Un subproceso puede ejecutar cualquier parte del código de un proceso, incluyendo partes que actualmente están siendo ejecutadas por otro subproceso.

11.1.2 *Acerca de los Procesos y los Subprocesos*

Cada proceso proporciona los recursos necesarios para ejecutar un programa. Un proceso tiene un espacio de dirección virtual, código ejecutable, referencias a objetos, variables de entorno, una prioridad base, y un mínimo y un máximo de espacio de trabajo. El espacio de trabajo es la memoria física asignada por el sistema operativo para el proceso. El espacio de trabajo contiene el código y las páginas de datos recientemente usadas por el proceso. Cada proceso es empezado con un subproceso solo, a menudo llamado subproceso primario, pero puede crear subprocesos adicionales desde cualquiera de sus subprocesos.

Todos los subprocesos de un proceso comparten su espacio de dirección virtual y los recursos del sistema. Además, cada uno mantiene control de excepciones, una prioridad de programa, y una serie de estructuras que el

sistema usará para guardar el contexto del subproceso hasta que sea fijado. El contexto del subproceso incluye la serie de subprocesos de los registros de la máquina, la pila del núcleo, un bloque de información de subproceso o Thread Information Block (TIB), y la pila del usuario en el espacio de direcciones del proceso del subproceso.

11.2 Métodos “Investigación y Desarrollo” de la Informática

11.2.1 *Las Diversas Clases de Métodos de Investigación*

Los métodos de investigación se pueden clasificar de acuerdo con distintos criterios, es necesario destacar que no existe consenso en la clasificación de los métodos de investigación. La diversidad de clasificaciones queda de manifiesto al comparar las clasificaciones propuestas por distintos autores (Atwater y Barbaria, 2004), (Gittins, 2002), (Jacobsen, 2004), (Zelkowitz & Wallace, 2004), entre otros, que se han ocupado del tema.

La clasificación que se presenta, en este apartado, responde principalmente a la postura que adopta el informático al investigar fenómenos o al abordar problemas reales o virtuales. Se aclara que los criterios utilizados para esta clasificación no son mutuamente excluyentes.

Se pueden establecer dos grandes clases de métodos de investigación: los métodos lógicos y los empíricos (figura 36). Los primeros son todos aquellos que se basan en la utilización del pensamiento en sus funciones de

deducción, análisis y síntesis, mientras que los métodos empíricos, se aproximan al conocimiento del objeto mediante sus conocimiento directo y el uso de la experiencia, entre ellos se encuentran, por ejemplo, la observación y la experimentación.

11.2.2 *Métodos Cuantitativos*

Son aquellos en los que se recogen y analizan datos cuantitativos sobre variables. La investigación cuantitativa trata de determinar la fuerza de asociación o correlación entre variables, la generalización y objetivación de los resultados a través de distintas técnicas (Gittins, 2002). Tras el estudio de la asociación o correlación pretende, a su vez, hacer inferencia causal que explique por qué las cosas suceden o no de una forma determinada.

Concibe el objeto de estudio como “externo” en un intento de lograr la máxima objetividad. Su concepción de la realidad social coincide con la perspectiva positivista. Es una investigación normativa, cuyo objetivo está en conseguir leyes generales referidas al grupo.

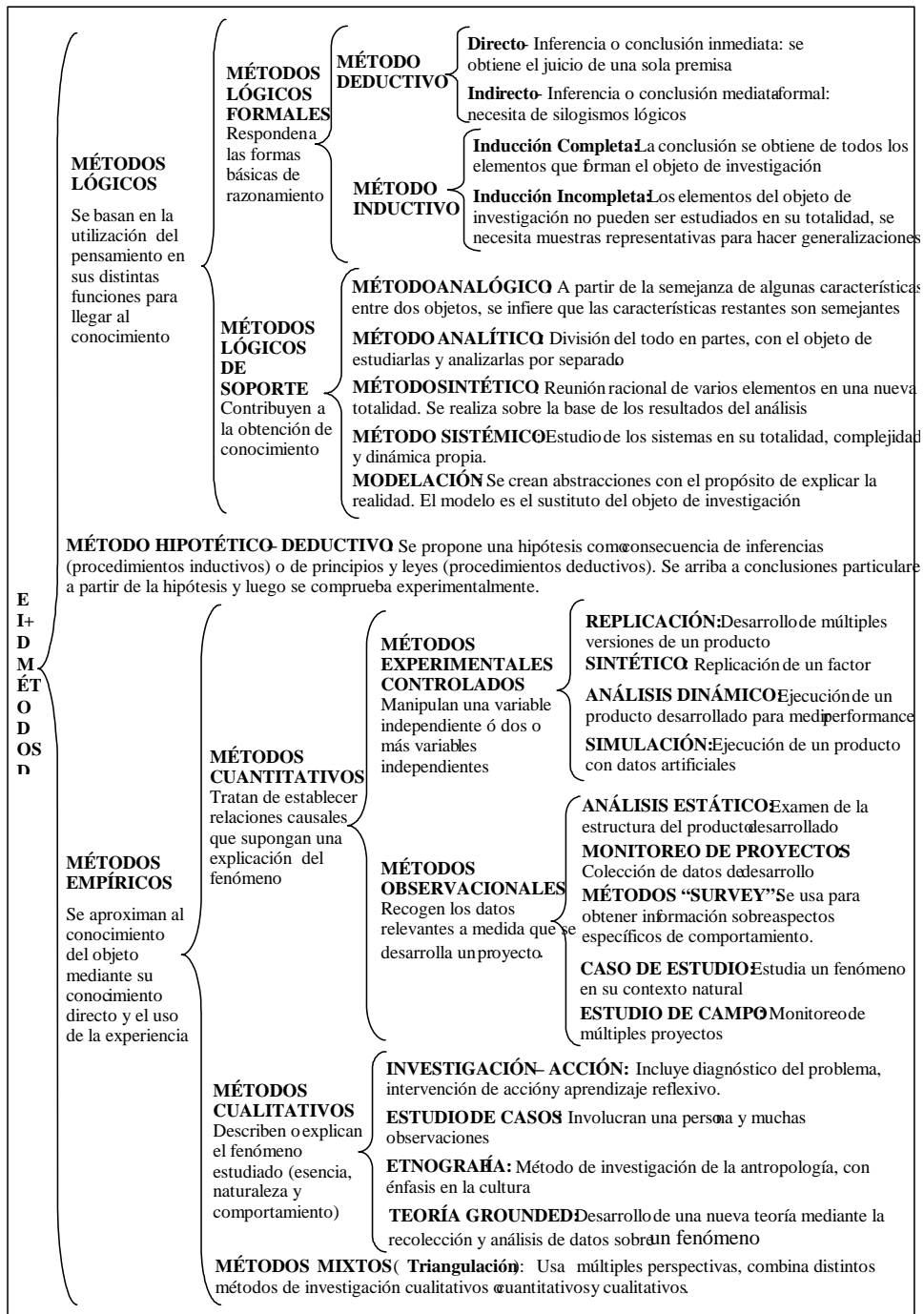


Figura. Nº36 Métodos de Investigación y Desarrollo

11.3 Componentes y Símbolos Utilizados en un Diagrama de Flujos de Datos

11.3.1 Componentes en un DFD

Estos componentes son las entidades que interactúan con el sistema, los almacenes de datos, los procesos involucrados y los flujos de información.

Entidades:

Proporcionan o usan datos del sistema.

Almacén:

Es un depósito de datos. Los procesos pueden introducir o recuperar datos de ellos.

En la documentación de un almacén de datos se debe incluir: nombre, descripción, contenido, flujos de entrada, flujos de salida y crecimiento.

Procesos:

Muestran lo que hace el sistema. Cada proceso tiene una o mas entradas de datos y produce una o mas salidas de información.

En la documentación de los procesos involucrados se debe incluir: nombre, descripción, flujos de entrada, flujos de salida y algoritmo (si corresponde).

Flujos de datos:

Modelan los movimientos de información en el sistema y se representan por líneas que unen los componentes. La dirección del flujo se indica con una flecha y la línea se etiqueta con el nombre del flujo.

En la documentación de los flujos de datos entre procesos se debe incluir: nombre, contenido, descripción, proceso origen, proceso destino y alias (si corresponde)

11.3.2 Símbolos en un DFD

Los símbolos utilizados en el diagrama de flujo de datos son los siguientes:

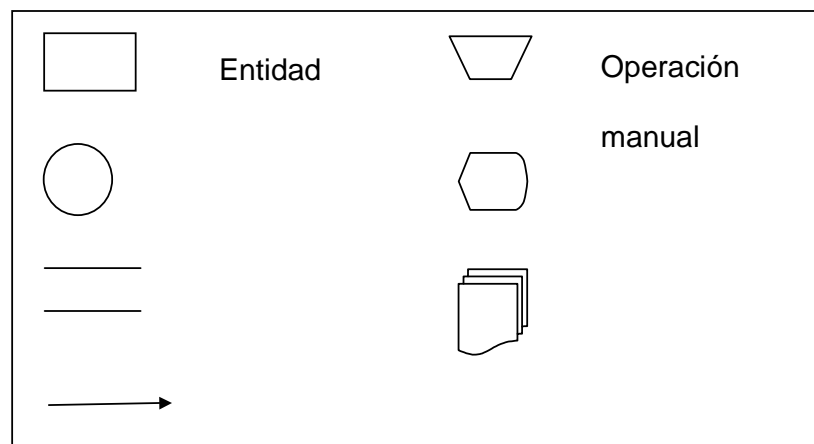


Figura. N°37 Símbolos utilizados en diagramas de flujos de datos.

11.3.3 *Nomenclatura en un diccionario de datos.*

La nomenclatura utilizada en la representación del diccionario de datos es la siguiente

'≡'	:	Está compuesto de
'.'	:	Y
'()'	:	Optativo (puede estar presente o ausente)
'{}'	:	Iteración (cero o mas ocurrencias)
'[]'	:	Seleccionar una de varias alternativas
'* *'	:	Comentario
'@'	:	Identificador (campo clave) para un almacén
' '	:	Separa opciones alternativas en la construcción
x{r}y	:	Límites para una iteración: x es el límite inferior, y es el limite superior.
'alias'	:	Indica que el flujo de datos puede tener otros nombres dentro del diccionario.

11.4 Descripción de Flujos de Datos del Diagrama de Contexto

datos_privilegios = Codtipo+nomtipo

Información de los privilegios de usuario

Proceso destino : Sistema para el control de procesos de aplicaciones Win32

datos_pc = mac+ip+nombrep+obs

Información de los computadores personales

Proceso destino : Sistema para el control de procesos de aplicaciones Win32

datos_usuario = rut+nombres+apellidos

Información de los usuarios que pueden administrar la aplicación

Proceso destino : Sistema para el control de procesos de aplicaciones Win32

datos_procesos_validos = nomproces+nameprogm+pathprocess+tamano

Información de los procesos válidos

Proceso destino : Sistema para el control de procesos de aplicaciones Win32

listado_usuarios = 1{rut+nombres+apellidos}

Informe de salida del sistema que representa la lista de usuarios que pueden administrar la aplicación.

Proceso destino : Sistema para el control de procesos de aplicaciones Win32

listado_procesos_validos = 1{nomproces+nameprogm+pathprocess+
tamano}

Informe de salida del sistema que representa la lista de procesos válidos. Este informe será mostrado por pantalla.

Proceso origen : Sistema para el control de procesos de aplicaciones Win32

listado_procesos_pc = 1{ nomproces+nombrep pc +mac }

Informe de salida del sistema que representa la lista de procesos válidos por pc. Este informe será mostrado por pantalla.

Proceso origen : Sistema para el control de procesos de aplicaciones Win32

11.5 Descripción de Simbología Utilizada en Diagrama Entidad

Relación

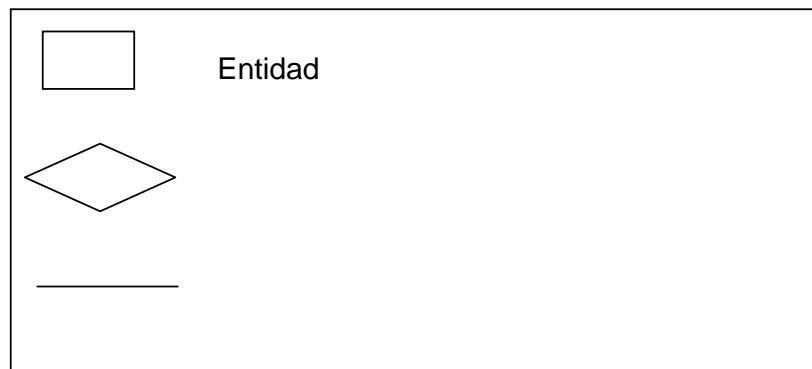


Figura. N°38: Simbología utilizada en diagrama E-R