

Universidad Nacional de Ingeniería

Facultad de Ingeniería Industrial y de Sistemas



Aplicación de la Ingeniería Reserva a un Sistema de Gestión Docente

Informe de Ingeniería

PARA OPTAR EL TÍTULO PROFESIONAL DE
INGENIERO DE SISTEMAS

Wilder Flores Ojeda

Lima - Perú

1994

CONTENIDO

| | |
|--|----|
| INTRODUCCION..... | 1 |
| OBJETIVOS..... | 3 |
| CAPITULO I | |
| 1.0 INGENIERIA REVERSA | 4 |
| 1.1 Introducción..... | 4 |
| 1.2 Orígenes de Ingeniería Reversa..... | 5 |
| 1.3 Definición y justificación..... | 6 |
| 1.3.1 Qué es Ingeniería Reversa en Ingeniería de Software?..... | 6 |
| 1.3.2 Qué se está tratando de efectuar con Ingeniería Reversa (Objetivos, Propósitos y Logros)?..... | 9 |
| 1.3.3 Cuando es necesaria Ingeniería Reversa?..... | 11 |
| 1.4 Fases de Ingeniería Reversa..... | 13 |
| 1.4.1 Fases de la planificación..... | 13 |
| 1.4.1.1 Analizar los objetivos del proyecto..... | 15 |
| 1.4.1.2 Inventario de componentes existentes..... | 16 |
| 1.4.1.3 Definir la estrategia de Ingeniería Reversa..... | 18 |
| 1.4.1.4 Decisión..... | 20 |
| 1.4.2 Fase de ejecución de las actividades de Ingeniería Reversa..... | 21 |
| 1.4.3 Fase de la consolidación/verificación de resultados..... | 22 |
| 1.5 Conclusiones..... | 23 |

CAPITULO II

| | |
|---|----|
| 2.0 TAXONOMIA DE CONCEPTOS RELATIVOS A INGENIERIA REVERSA..... | 24 |
| 2.1 Mantenición de Software..... | 24 |
| 2.2 Ciclo de vida y abstracciones..... | 24 |
| 2.3 Reusabilidad de software..... | 25 |
| 2.3.1 Tipos de reuso..... | 26 |
| 2.3.2 Tecnología de reuso y recuperación de componentes..... | 26 |
| 2.3.3 Beneficios de reuso..... | 27 |
| 2.4 Rejuvenecimiento de software..... | 27 |
| 2.4.1 Redocumentación..... | 29 |
| 2.4.1.1 Recuperación de Diseño..... | 31 |
| 2.4.1.1.1 Etapas del proceso de Recuperación de Diseño..... | 31 |
| 2.4.2 Restructuración..... | 34 |
| 2.4.2.1 Formas de Restructuración..... | 36 |
| 2.4.2.2 La promesa de Restructuración..... | 36 |
| 2.4.2.3 Los problemas de Restructuración..... | 36 |
| 2.4.3 Reingeniería..... | 37 |
| 2.4.3.1 Introducción a la transformación de código fuente en la Reingeniería de software... | 39 |
| 2.4.3.2 Complejidad y costo de mantención..... | 40 |
| 2.4.3.3 Reingeniería de procedimientos..... | 40 |
| 2.4.3.3.1 Reestructuración de programas... | 40 |
| 2.4.3.3.2 Eliminación de código redundante | 41 |
| 2.4.3.3.3 Eliminación de código malo..... | 41 |
| 2.4.3.3.4 Remodularización..... | 41 |
| 2.4.3.3.5 Traducción de lenguajes..... | 41 |
| 2.4.3.4 Reingeniería de datos..... | 42 |
| 2.4.3.4.1 Problemas de la definición de datos..... | 42 |
| 2.4.3.4.2 Problemas de los valores de datos | 42 |
| 2.4.3.4.3 Reingeniería de datos: El proceso..... | 42 |
| 2.4.3.4.4 Racionalización de nombres de datos..... | 43 |

| | |
|--|----|
| 2.4.3.4.5 Reestructuración de datos..... | 43 |
| 2.4.3.4.6 Normalización de datos..... | 43 |
| 2.4.4 Conclusiones y visión futura..... | 44 |

CAPITULO III

| | |
|--|----|
| 3.0 APLICACIONES DE INGENIERIA REVERSA AL SISTEMA DE GESTION DOCENTE..... | 46 |
| 3.1 Introduucción..... | 46 |
| 3.2 Objetivos..... | 47 |
| 3.3 Descripción del Sistema de Gestión Docente..... | 47 |
| 3.3.1 Régimen Académico..... | 47 |
| 3.3.1.1 Estructura Académica..... | 48 |
| 3.3.1.2 Docentes Universitarios..... | 49 |
| 3.3.1.2.1 Docentes Ordinarios | 50 |
| 3.3.1.2.2 Requisitos para cada categoria de Profesores Ordinarios..... | 51 |
| 3.3.1.2.3 Docentes Extraordinarios..... | 52 |
| 3.3.1.2.4 Docentes Contratados..... | 53 |
| 3.3.1.2.5 Jefes de Prácticas..... | 54 |
| 3.3.1.3 Reglamento General de Docentes..... | 55 |
| 3.3.1.3.1 Evaluación, Promoción y Ratifica- ción de Docentes..... | 55 |
| 3.3.1.3.2 Deberes y derechos de los Docentes | 56 |
| 3.3.1.3.3 Sanciones..... | 57 |
| 3.3.1.3.4 Incompatibilidades y pérdida de la dedicación exclusiva y tiempo com- pleto..... | 58 |
| 3.3.2 Carga Académica Docente..... | 58 |
| 3.3.2.1 Criterios de Asignación de Carga Aacadémica | 58 |
| 3.3.2.2 Carga Lectiva..... | 59 |
| 3.4 Estudio preliminar del problema..... | 60 |
| 3.5 Proceso de redocumentación..... | 65 |
| 3.6 Proceso de Ingeniería Reversa..... | 65 |
| 3.6.1 Etapa de planificación..... | 66 |
| 3.6.1.1 Analizar los objetivos del proyecto..... | 67 |
| 3.6.1.2 Inventario de componentes existentes.... | 68 |

| | |
|--|------------|
| 3.6.1.3 Definir la estrategia de Ingeniería Reversa..... | 72 |
| 3.6.2 Etapa de ejecución de las actividades de Ingeniería Reversa..... | 76 |
| 3.6.2.1 Diagrama de Entidad/Relación..... | 76 |
| 3.6.2.2 Diagrama de Flujo de Datos..... | 78 |
| 3.6.2.3 Diagramas de Descripción de Archivos.... | 80 |
| 3.6.2.4 Arbol de Menúes..... | 82 |
| 3.6.3 Etapa de consolidación y verificación de los resultados..... | 83 |
| 3.7 Documentación del Sistema: Gestión Docente..... | 83 |
| 3.7.1 Diagrama de Entidad-Relación..... | 85 |
| 3.7.2 Diagrama de Contexto..... | 88 |
| 3.7.2.1 Diagrama de Primer Nivel..... | 89 |
| 3.7.2.2 Explosión de Burbujas..... | 90 |
| 3.7.3 Diagramas y Reglas de Procesos..... | 94 |
| 3.7.4 Descripción de Archivos..... | 109 |
| 3.7.5 Arbol de Menues..... | 122 |
| 3.7.6 Tablas de Procesos versus Archivos..... | 123 |
| CONCLUSIONES Y RECOMENDACIONES..... | 124 |

Anexo 1: INGENIERIA REVERSA Y LAS LEYES DE PROPIEDAD INTELECTUAL

| | |
|---|-----|
| 1.1 Introducción..... | 127 |
| 1.2 Interpretando las Leyes..... | 127 |
| 1.3 El Uso Legal..... | 128 |
| 1.4 Los Secretos de Negocios..... | 128 |
| 1.5 Que se puede realizar..... | 128 |
| 1.5.1 Hacer modificaciones..... | 129 |
| 1.5.2 Desarrollar un Software similar..... | 129 |
| 1.6 Diferentes Leyes en Ingeniería Reversa..... | 129 |

Anexo 2: ASPECTOS GENERALES DE LA HERRAMIENTA INGRES

| | |
|---|-----|
| 2.1 Introducción..... | 131 |
| 2.2 Qué es INGRES ?..... | 132 |
| 2.3 Descripción técnica del Software..... | 132 |

| | |
|---|-----|
| 2.3.1 Estadares..... | 132 |
| 2.3.2 Portabilidad de las aplicaciones..... | 133 |
| 2.3.3 Utilitarios de Administración de Información.. | 133 |
| 2.3.4 Interfaz a lenguajes de programación..... | 134 |
| 2.3.5 Herramientas para el desarrollo de aplicaciones..... | 134 |

| | |
|--------------------------|------------|
| BIBLIOGRAFIA..... | 135 |
|--------------------------|------------|

INTRODUCCION

Este proyecto nace de la necesidad de buscar una metodología que permitiese desarrollar una serie de actividades, para atacar aquellos problemas informáticos que están basados sobre un sistema de software existente, para lo cual surgió el nombre de Ingeniería Reversa. Una metodología nueva que se está recién incorporando a los productos de software. Este proyecto se inicia investigando, analizando y profundizando esta metodología, para luego aplicarla a un Sistema de Información existente.

El primer capítulo detalla la teoría sobre la cual se basa la metodología de Ingeniería Reversa. se define cuando y como utilizar esta metodología dentro de la mantención de sistemas. Además, se analizan y detallan las diferentes actividades involucradas en las fases en las cuales está estructurada esta metodología en estudio, para con esto lograr un acabado y completo entendimiento de la función y aplicación de ésta.

El segundo capítulo muestra una taxonomía de métodos relativos a Ingeniería Reversa, en forma comparativa con esta última metodología. Estos métodos relativos se encuentran todos clasificados dentro del mismo ámbito de la metodología de estudio, es decir, el Rejuvenecimiento de Software.

En el tercer capítulo se detalla la aplicación de la metodología en estudio sobre el Sistema de Gestión Docente existente en la Universidad Andina de Juliaca (sistema en

operación), con el objetivo de permitir un mejor entendimiento y examinación de este. Para luego migrar a un ambiente nuevo y contar con documentación actualizada.

Para finalizar este proyecto se incluye un anexo de las implicancias legales que tiene la metodología en estudio en su utilización, con respecto a los derechos de propiedad intelectual. También se incluyen anexos con respecto a los aspectos generales de la herramienta Ingres y la documentación del sistema de Gestión Docente que resultó de la aplicación de la metodología en cuestión.

OBJETIVOS

Objetivos Principal

Conocer y aplicar la metodología de Ingeniería Reversa en el área de Ingeniería de Software.

Objetivos específicos

- Obtener documentación y experiencia sobre la aplicación de Ingeniería Reversa en Sistemas de Información.
- Lograr con la metodología de Ingeniería Reversa un suficiente entendimiento del Sistema de Información Gestión Docente existente en la Universidad Andina de Juliaca.
- Migrar el Sistema de Gestión Docente al ambiente de desarrollo de sistemas "Ingres", ya que la Universidad de Carolina de Norte Lenoir-Rhyne College donó un equipo Digital Vax/VMS con el DBMS en mención.

CAPITULO I : INGENIERIA

REVERSA

1.1 INTRODUCCION

Se estima que la mayoría de los productos de software, están mal o escasamente documentados y su mantención requiere de más de la mitad del presupuesto software de los departamentos de informática. Además, la inversión en ese software es tan alta y la demanda por nuevos sistemas es tan grande, que no hay tiempo ni presupuesto para rehacer todo desde cero. Para agravar la situación, la tarea de mantención de software aparece como poca atractiva, no existe una conciencia de mantención es decir, los analistas y programadores prefieren crear software antes que mantener el desarrollo de otros.

Este gran problema de la mantención de software está siendo abordado exitosamente por una metodología denominada Ingeniería Reversa. El objetivo de esta metodología es obtener o ganar un suficiente entendimiento del nivel de diseño para ayudar a la mantención. En Ingeniería de Software, la idea es analizar un sistema representado como una especificación, un diseño o simplemente un trozo de código, con el objeto de generar una nueva representación, de más alto nivel o bien del mismo nivel, pero en una forma más entendible.

Las herramientas de Ingeniería Reversa han sido concebidas no para modificar un sistema, sino para comprenderlo y posiblemente obtener una versión funcionalmente idéntica pero de mejor calidad. Con frecuencia, una empresa toma sus sistemas antiguos y mal documentados, pero en operación, para modernizarlos y adecuarlos a sus nuevos requerimientos de información y de servicio.

Normalmente, quien utiliza métodos de Ingeniería Reversa sobre software no es el creador original de éste. Esto tiene algunas implicancias legales, ya que se pueden infringir las leyes de propiedad intelectual.

1.2 ORIGENES DE INGENIERIA REVERSA

El término Ingeniería Reversa tiene sus orígenes en el análisis de hardware donde se practica descifrando el diseño de productos terminados. Ingeniería Reversa es aplicada regularmente para optimizar o mejorar los propios productos y también para analizar los productos del competidor.

Ingeniería Reversa se define como "el proceso de desarrollo de un conjunto de especificaciones para un complejo sistema de hardware, con una ordenada examinación de piezas de este sistema".

En las aplicaciones de este concepto para sistemas de software, se encuentra que muchas de éstas se aplican para ganar un entendimiento básico de un sistema y de su estructura. Sin embargo, mientras el objetivo en el caso del hardware es para duplicar el sistema, el objetivo para el software es frecuentemente el ganar un suficiente entendimiento del nivel de diseño para ayudar a la tarea de mantención.

1.3 DEFINICION Y JUSTIFICACION

1.3.1 ¿ QUE ES INGENIERIA REVERSA EN INGENIERIA DE SOFTWARE ?

Ingeniería Reversa es un proceso de extracción y recuperación de abstracciones e información de diseño (documentación) del sistema existente. Esta información obtenida puede ser utilizada para posteriores desarrollos, mantenciones, Reingenierías, o propósitos de Reusos (ver figura 1.1).

En el diagrama de flujo de datos del proceso de Ingeniería Reversa (Figura 1.1), participan una serie de entidades que es necesario detallar:

- **Mercado** : Es el ambiente de marketing donde se mueve los sistemas de software. Estos salen al mercado de acuerdo a las demandas o necesidades de los consumidores.
- **Usuario** : Es la persona u organización que directamente trabaja con los sistemas de software. Es quien entrega los requerimientos o necesidades a satisfacer por el sistema de acuerdo a los objetivos fijados.
- **Ambiente de Procesamiento de Datos** : Lugar donde se evalúa el desempeño del sistema y se generan resultados que conducirán a tomar decisiones con respecto a su comportamiento y a las necesidades de modificación.
- **Operaciones** : Refleja el comportamiento del sistema implementado en su etapa de operación y entrega los problemas operativos que se detectan.
- **Diseño / Rediseño** : Representa el modelamiento de un sistema que desea implementar o de un sistema existente (sistema ya implementado).

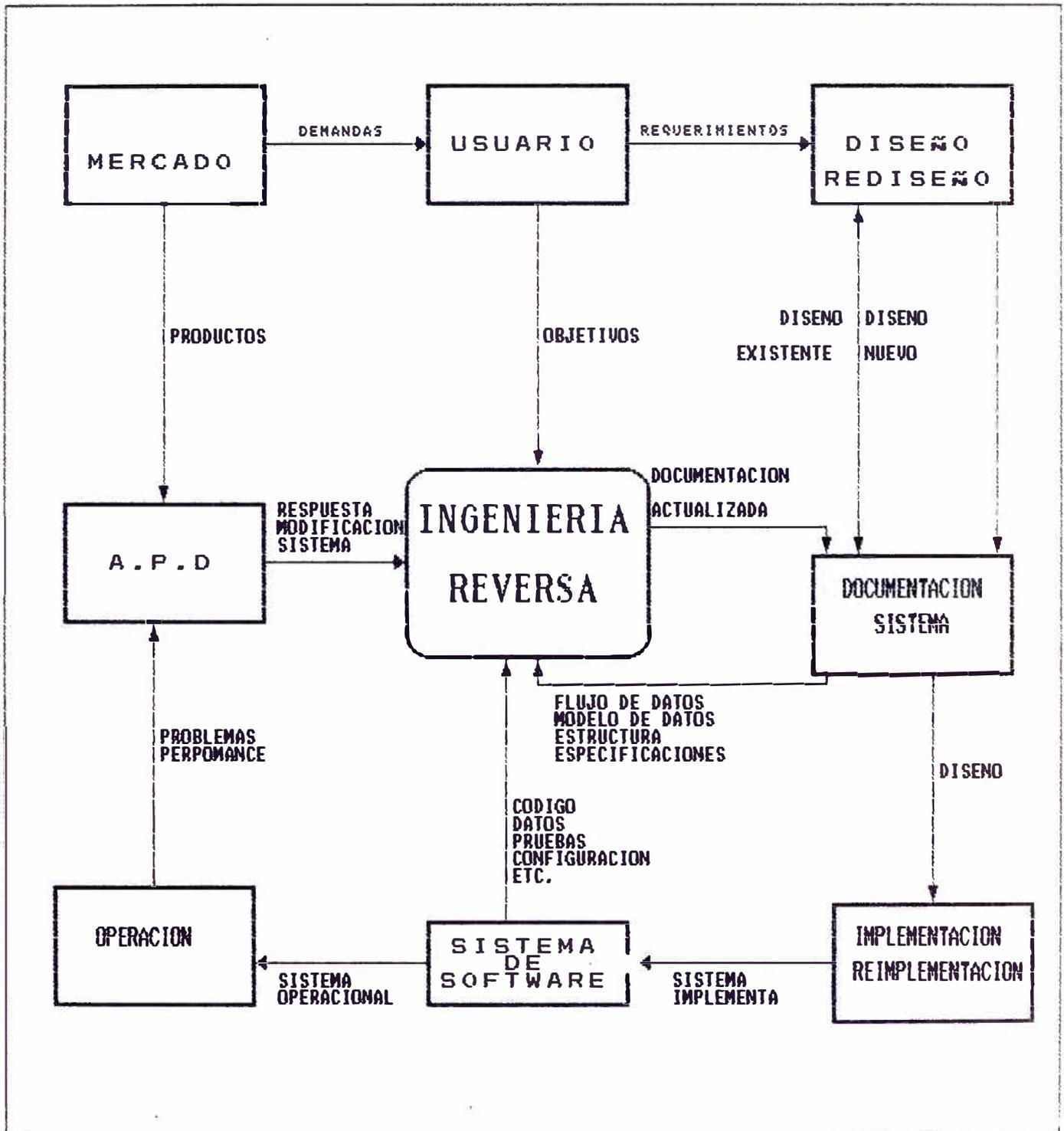


FIGURA 1.1 Diagrama de flujo de datos de ingeniería inversa

- **Implementación / Reimplementación** : Representa la etapa de construcción de un sistema, éste puede ser un sistema nuevo o un sistema existente que se desea redesarrollar (recodificar).

Los archivos que se incluyen en el diagrama de flujo de datos representan la **documentación del sistema** (son todos los documentos o información que se mantiene del sistema implementado) y sistema de software (es el código y los datos propios de la implementación y operación respectivamente del sistema implementado).

Todas las entidades antes mencionadas participan directa o indirectamente en el proceso de Ingeniería Reversa, el cual de acuerdo a los objetivos planteados y los resultados del desempeño operacional del sistema se logrará una optimización de la documentación del sistema y a su vez se obtendrá un rápido y mejor entendimiento del sistema con un eficiente análisis de reusabilidad, para un posible futuro Rediseño y/o Reimplementación del sistema en estudio, de acuerdo a los objetivos y necesidades del usuario.

Ingeniería Reversa es el proceso de análisis de un sistema existente, cuyos objetivos son los siguientes:

- Identificar los componentes del sistema y su interrelación.
- Crear representaciones del sistema con un alto nivel de abstracción.

Esta metodología generalmente no realiza cambios al sistema, es mas un proceso de examinación y no un proceso de cambio o réplica.

En el ciclo de vida de los sistemas de información, Ingeniería Reversa cubre un amplio rango comenzando de la implementación existente, capturando el diseño y descifrando los requerimientos actualmente implementados por el sistema.

Ingeniería Reversa va más allá de la simple redocumentación, emprende una recuperación de la información basada sobre la especificación y métodos de diseño de software y su posterior almacenamiento de esta información en un depósito para ser manipulada por un mantenedor de software.

El proceso de Ingeniería Reversa es ilustrado en la figura 1.2., obtiene representaciones de alto nivel de abstracción como lo son los diccionarios de datos, modelos de datos, flujos de datos, etc.. El método de análisis y diseño estructurado estándar sirve como un buen mecanismo de comunicación para articular la información de Ingeniería Reversa, es decir, diccionarios de datos, flujos de datos, flujos de control, modelo de datos, etc..

1.3.2. ¿ QUE SE ESTA TRATANDO DE EFECTUAR CON INGENIERIA REVERSA (OBJETIVOS, PROPOSITOS Y LOGROS) ?

El Primer propósito en un sistema de software es lograr una óptima comprensión del sistema para las posibles mantenciones y los nuevos desarrollos que se puedan realizar en el futuro.

Más allá de la definición de Ingeniería Reversa, hay una serie de objetivos claves que guiarán esta tecnología.

- **Combatir la complejidad.** Se deben desarrollar métodos para disminuir la complejidad del sistema. Una clave para controlar estos atributos es automatizar el apoyo en el desarrollo de sistema. Los métodos y herramientas de Ingeniería Reversa con ambiente Case proveerán una forma para extraer información relevante, para tomar decisiones que puedan controlar el proceso y el producto en la evolución del sistema.

- **Generar vistas alternativas.** Las representaciones gráficas han sido largamente aceptadas como una ayuda en la comprensión

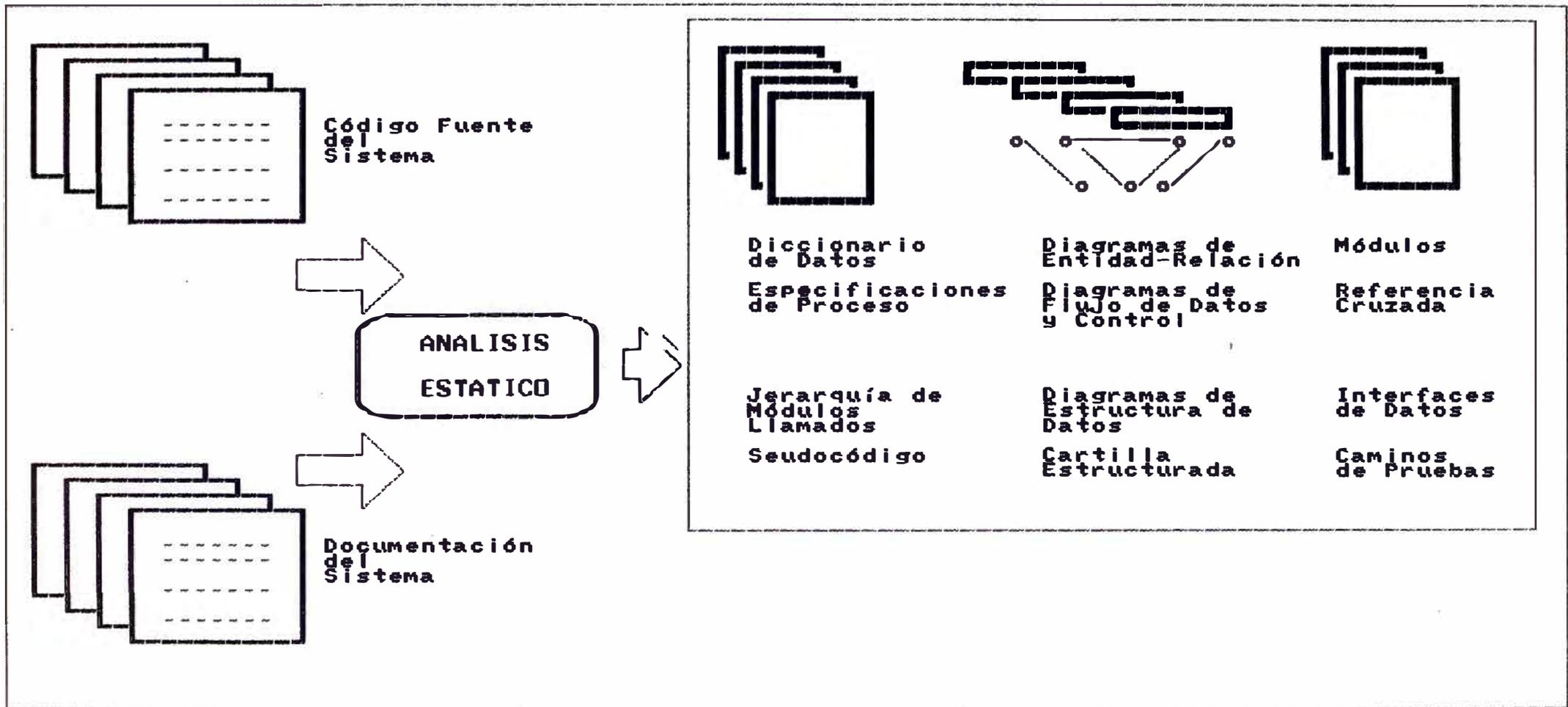


Figura 1.2 Proceso de Ingeniería Reversa

de los sistemas de software. Sin embargo, la mantención y creación de ellas continua siendo un obstáculo en el proceso. Las herramientas de Ingeniería Reversa facilitan la generación o regeneración de representaciones gráficas de otras formas. Mientras muchos diseñadores trabajan en una singular perspectiva primaria (ejemplo, diagrama de flujo de datos), las herramientas de Ingeniería Reversa pueden generar vistas adicionales de otras perspectivas (ejemplos, diagramas de flujo control, cartillas estructuradas, diagramas entidad-relación. etc.) para ayudar al proceso de revisión y verificación.

- **Recuperar la información perdida.** La continua evolución de un sistema de larga vida conduce a la pérdida de información sobre el diseño del sistema. Las modificaciones frecuentemente no son reflejadas en la documentación, particularmente a un alto nivel de abstracción. Ingeniería Reversa recupera las especificaciones de Diseño que es una forma para solucionar el problema anteriormente mencionado.

- **Facilitar el reuso.** Ingeniería de Reversa ayuda a detectar candidatos para componentes de software reusables del sistema existente.

Ingeniería de Reversa ha llegado a ser rápidamente una reconocida e importante componente de futuros ambientes Case. Las herramientas de Ingeniería Reversa pueden proveer un mejor enlace en las etapas del ciclo de vida de desarrollo y mantención. Además, usadas con tecnología de desarrollo de software, proveen un significativo incremento en la productividad de software.

1.3.3. ¿ CUANDO ES NECESARIA INGENIERIA REVERSA ?

Es claro que esta metodología rara vez será justificada económicamente, sino asociada con otros objetivos orientados a

la productividad. Usualmente la necesidad de esta metodología nace de los problemas operacionales reales del sistema existente, semejante a:

- Errores y caídas del sistema ocurridas por una mala calidad del software.
- Alto costo de mantención del software que ocurren por la carencia de estructura, transparencia y documentación.
- Cambios en la infraestructura hardware o software, requiriendo por lo tanto de una migración o conversión.
- Nuevos requerimientos de los usuarios.
- Requerimientos de seguridad para reducir los riesgos en las aplicaciones.
- Extender la funcionalidad de una aplicación con problemas de documentación y/o diseño.

Todos estos problemas pueden ser resueltos por actividades, tales como, Reingeniería, Rediseño (Reestructuración), conversión migración, mantención preventiva, etc.. Las cuales todas requieren al comienzo un buen y confiable entendimiento del sistema existente.

Ingeniería Reversa puede solamente ser justificada cuando sus costos son menores que los costos de redesarrollo o por un producto estándar. Pero esta decisión puede solamente ser tomada, después que ambas alternativas han sido evaluadas en su amplia dimensión (costos directos e indirectos y beneficios).

Ingeniería Reversa ha llegado a ser una atractiva inversión, porque ésta da una garantía, que el trabajo ahora realizado será fácilmente reutilizado en el futuro.

1.4 FASES DE INGENIERIA REVERSA

La entrada a esta metodología consiste de todo el material o información disponible relativa al desarrollo, operación y mantención del sistema existente. El resultado es un completo entendimiento y documentación actualizada, conteniendo toda la información necesaria para una eficiente y confiable utilización, operación o modificación del sistema.

Esta metodología está estructurada en tres fases principales, las cuales se realizan en forma secuencial (ver figura 1.3) :

1.0.- Planificación

2.0.- Ejecución de las actividades

3.0.- Consolidación / Verificación de los resultados

1.4.1 FASES DE LA PLANIFICACION

El plan de trabajo puede solamente ser elaborado después de un primer y modesto análisis cuantitativo y cualitativo del material existente (calidad del sistema existente) para encontrar una estrategia óptima para la realización del trabajo de Ingeniería Reversa.

Esta fase describe el rol de cada componente del sistema existente, como también los recursos y herramientas requeridas, y la secuencia de acciones a ser realizadas.

Además se identifican aquellos componentes que no satisfacen los requerimientos de la nueva necesidad, porque ellos no son relevantes a la aplicación, o porque el redesarrollo o cambio por un producto comercial son una alternativa más óptima.

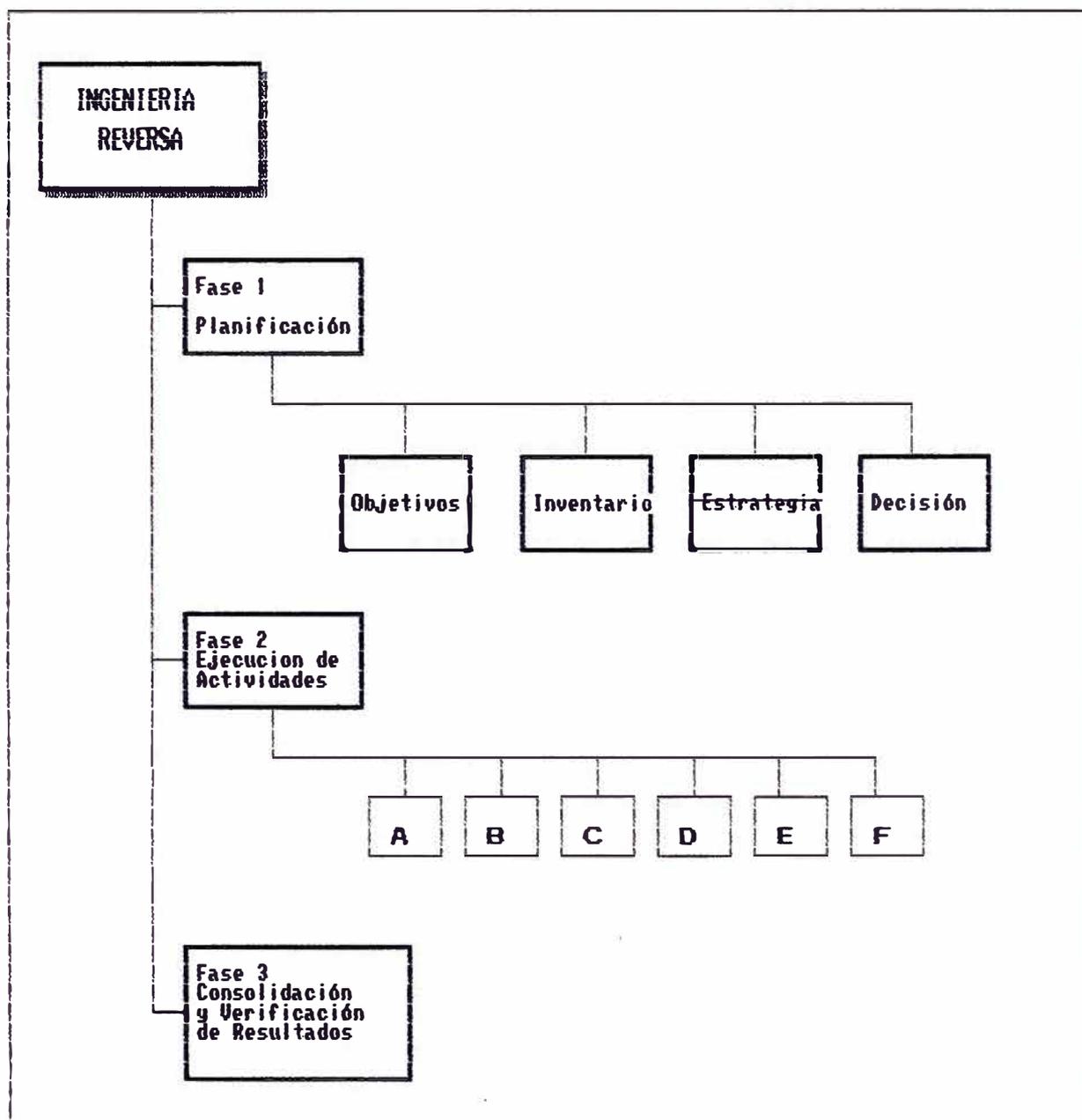


Figura 1.3 Etapas del proceso Ingeniería Reversa

Esta fase a su vez se divide en otras actividades:

- 1.0.- Analizar los **objetivos** del proyecto
- 2.0.- **Inventario** de los componentes existentes
- 3.0.- Definir la **estrategia** de Ingeniería Reversa
- 4.0.- **Decisión**

1.4.1.1 ANALIZAR LOS OBJETIVOS DEL PROYECTO

El primer paso en la preparación de cualquier plan, es estar seguro que los objetivos del proyecto están claramente expuestos para que sean aprobados por las instancias de decisión.

Como lo antes expuesto, Ingeniería Reversa es rara vez un objetivo en sí mismo. La usual justificación para la metodología antes mencionada es que algunos problemas en el sistema existente han de ser resueltos y lograr una calidad global en el futuro sistema, sin tener que realizar un proceso de redesarrollo.

En particular:

- La documentación podrían ser una exacta y concisa representación de la implementación de un sistema, permitiendo a los analistas que la requieran usar, operar, mantener o modificar ésta y también podría utilizarse para evaluar el sistema, para lograr familiarizarse rápidamente con sus características y arquitectura.
- La conformidad del sistema para los requerimientos del usuario deben haber sido probados y verificados, y esta confianza podría ser medida en cualquier momento.

La clave para lograr los objetivos es tener un modelo de la información necesaria para el diseño, implementación y operación del sistema de información. Tales modelos (meta-modelos) son relativamente independientes del tipo de sistema de información considerando.

El concepto meta-modelo y depósito va más de acuerdo con ambientes de desarrollo de sistema Case. Para el proceso manual se utiliza solamente la teoría del concepto depósito, en otras palabras, el término depósito indica en forma abstracta el lugar donde se mantiene toda la información o documentación del sistema en cuestión.

En síntesis, en esta fase se incluirán sentencias sobre :

- La justificación original para la fase
- La naturaleza de los resultados
- El meta-modelo del depósito
- Los métodos y herramientas (Case) usadas para presentar estos resultados.

1.4.1.2 INVENTARIO DE COMPONENTES EXISTENTES

Después de tener claramente definidos los objetivos, la dimensión y calidad del sistema existente, esta información debe ser evaluada. La mejor forma para hacer esto, es hacer un inventario de todos los componentes disponibles relativos al sistema existente.

Al más bajo nivel, un componente elemental es un objeto singular (archivo o documento), el cual debe tener un único nombre. Pero los componentes pueden ser también una colección de componentes de más bajo nivel, y el sistema mismo completo también es un componente.

Es importante hacer notar que en esta representación es necesaria solamente una lista de todos los componentes disponibles o componentes relativos al sistema existente y no un detallado análisis de éstos.

Cada componente puede ser asignado a una clase correspondiente a los tipos de entidades del meta-modelo, en particular:

- Código fuente de los programas
- Archivos de documentación, manuales, etc.
- Archivos ejecutables y librerías de tiempo de ejecución
- Procedimientos organizacionales y requerimientos relativos a la utilización del sistema de información.
- Información administrativa relativa al desarrollo original y a la mantención del sistema existente (requerimientos, costos, estándares, etc.)
- Información de validación y aceptación (planes de prueba, resultados de prueba, informe de aceptación, informe de calidad y métricas)
- Tablas y parámetros de configuración. Estas tablas y parámetros son actualmente usadas para configurar el sistema hardware y software
- Archivos de datos y Bases de datos usadas por la operación del sistema, incluyendo interfases, informes y mensajes transferidos
- Componentes hardware y software

Obviamente, existen algunas dependencias entre estas clases, las cuales deben estar documentadas (por ejemplo, un componente ejecutable es generado por algún componente de código fuente, el cual está documentado en algún archivo de documentación).

Cada componente es identificado por un único nombre y una corta descripción, y es asociado a una de las clases antes definidas. Esta también contiene una referencia a la localización del componente.

Ya que el inventario delimita el alcance del área investigada, esto es también factor esencial para el completo éxito administrativo del proyecto de Ingeniería Reversa).

1.4.1.3 DEFINIR LA ESTRATEGIA DE INGENIERIA REVERSA

El próximo paso requiere un primer análisis global de la calidad y reusabilidad de cada componente identificado. Ya que el número de componentes inventariados puede ser muy grande, es importante procesar sistemáticamente y con apropiadas herramientas. Una aproximación a este problema está basado sobre los siguientes principios:

a) Jerarquía (aproximación Top_down)

La relación "es_parte_de" puede ser usada para construir este tipo de estructura jerárquica con los componentes. El límite de la profundidad del análisis, depende sobre el atributo global asignado al componente tope.

Un componente puede ser completamente descartado, o ser reemplazado por un producto estándar, el análisis puede terminar aquí, y en una primera aproximación todos los atributos pueden ser heredados a los componentes dependientes de niveles inferiores. En caso contrario, en orden a maximizar

el reuso, si el objeto va a ser modificado o completamente desarrollado, el análisis debe ser detallado hasta el nivel más bajo de los componentes.

b) Atributos de Calidad

Unos pocos atributos básicos de calidad son suficientes para decidir sobre la forma a procesar. Un componente puede ser manejado por Ingeniería Reversa de acuerdo a los atributos de: apoyo, estructura y relevancia. Una simple codificación de éstos atributos hace más fácil construir grupos de componentes con similares características y seleccionar la mejor estrategia para procesar cada uno de ellos.

i) Apoyo: Este atributo responde la consulta de si un componente está disponible en forma electrónica o no. Si un componente está disponible en forma electrónica puede ser procesado directamente por herramientas de software. En caso contrario, un componente disponible manualmente requerirá algún tipeo (o la utilización de un scanner) para que luego esté disponible en el depósito.

ii) Estructura: Este atributo evalúa la estructura de un componente: Un componente estructurado es informal y puede ser solamente analizado manualmente después de leer el documento completo. Un componente muy bien estructurado puede fácilmente ser interpretado por herramientas automáticas.

iii) Relevancia: Este atributo evalúa el grado de contribución de un componente a los objetivos y la calidad de un sistema de información. Un componente no relevante nunca es usado y puede ser eliminado del sistema sin afectar su funcionalidad y es esencial en la correcta operación del sistema, o por su calidad.

Para cada posible combinación de estas opciones, una de las siguientes tres estrategias básicas de Ingeniería

Reversa, puede ser aplicada a los correspondientes componentes inventariados:

- El componente **no es utilizable**, y será descartado; la estrategia de Ingeniería Reversa es reconstruir éste directamente en el depósito.
- El componente **satisface los objetivos de Ingeniería Reversa**, y puede ser tomado igual (sin modificaciones); solamente la acción requerida será transferir éste al depósito.
- El componente **contiene partes reusables**, las cuales pueden ser analizados y extraídas por herramientas automáticas o manualmente; la estrategia de Ingeniería Reversa especificará las herramientas y técnicas a ser usadas y los resultados esperados del análisis.

La lista de componentes estáticos pueden ser transformados en una lista de actividades a ser realizadas, junto con los recursos necesarios y herramientas requeridas. En algunos casos complejos puede ser útil dibujar un Diagrama de Flujo de Datos de estas actividades, para mostrar las dependencias lógicas entre ellas. Después de estimado el esfuerzo y el costo requerido por cada actividad, se evalúa el costo total del proyecto de Ingeniería Reversa y las posibles alternativas pueden ser determinadas aproximadamente. Luego puede ser tomada la decisión para proceder con la siguiente fase.

1.4.1.4 DECISION

El resultado final de la fase de planificación es un plan para la ejecución de las actividades de Ingeniería Reversa. Este provee una aproximada evaluación de los costos envueltos en varias alternativas, y permite al administrador tomar la decisión correcta. Sin embargo, puede resultar de éste análisis que al menos para algún componente del sistema,

Ingeniería Reserva no sea la más óptima para lograr los objetivos fijados, esto conducirá a la reducción en el alcance del proyecto. Pero el esfuerzo gastado en verificar y analizar el inventario, no puede ser considerado como una pérdida para la organización. El proyecto ya tendrá un resultado invaluable, el cual justifica los riesgos tomados al comienzo.

1.4.2 FASE DE EJECUCION DE LAS ACTIVIDADES DE INGENIERÍA REVERSA

Las actividades de Ingeniería reversa a ser realizadas son descritas en el plan elaborado en la etapa previa. Su naturaleza es fuertemente dependiente sobre el contexto del proyecto.

Básicamente, la principal tarea consiste en la manipulación de información almacenada en los archivos textos (documentos, manuales), y la transformación de esta información en algún dato estructurado definidos en el meta-modelo.

En el mejor de los casos, esta transformación puede ser hecha con las herramientas automáticas, pero en la práctica el proceso manual es requerido, esto va a depender de la calidad del material existente. Verdaderamente, las actividades de Ingeniería Reversa usualmente son una combinación de una o más de las siguientes tareas elementales:

[A] Mejorar la estructura de un documento apoyado por la Reestructuración manual.

[B] Mejorar la relevancia de un documento apoyado por una selección de las partes relevantes.

[C] Hacer disponibles documentos a través del tipeo o scanning, obviamente, esta operación es solamente

significativa para los documentos que ya tienen un nivel de aceptación en la estructura y relevancia.

[D] Mejorar la relevancia de documentos estructurados, a través de una selección usando un lenguaje de manipulación de datos interactivos o Base de Datos.

[E] Revisar y mejorar la estructura de diseño de la Base de Datos, eliminando redundancia para lograr una mejor estandarización con respecto al meta-modelo. Y además asegurar que ellos sean coherentes con la realidad.

[F] Transferir los datos de diseño al ambiente de documentación destino, la cual es, mayormente usada en ambientes de tecnología de objetos (depósito).

1.4.3 FASE DE CONSOLIDACION/VERIFICACION DE RESULTADOS

El propósito de la etapa anterior es complementar o actualizar el depósito con información extraída en forma automática o manual del material existente.

Para poder completar las fases de Ingeniería Reversa es ahora necesaria consolidar este diseño, verificando su totalidad y consistencia, y ampliando éste con apropiadas introducciones. Después, el depósito podría ser una fiel representación del actual sistema de información analizado y esto puede servir como una base confiable para cualquier futura Reingeniería, Rediseño, etc..

Los depósitos basados en herramientas de diseño (Case) son usualmente utilizadas para estas tareas.

Si se encuentran disponibles, adecuados generadores de código, conectados al depósito, es posible regenerar automáticamente partes del sistema existente con el diseño de

Ingeniería Reversa.

Los resultados podrían ser funcionalmente idénticos al sistema original.

**CAPITULO II : TAXONOMIA DE
CONCEPTOS RELATIVOS A INGENIERIA
REVERSA**

2.1 MANTENCION DE SOFTWARE

La definición de Mantenición de Software es "la modificación de un producto de software después de deliberadas correcciones de fallas, para mejorar el desempeño u otros atributos, o para adaptar el producto a cambios del ambiente."

Usualmente, los mantenedores de sistemas no son los diseñadores originales, por lo tanto, ellos deben gastar muchos recursos para examinar y conocer el sistema, Ingeniería Reversa puede facilitar esta práctica. En este contexto Ingeniería Reversa es la parte del proceso de mantención que ayuda al entendimiento del sistema, para que se pueda realizar apropiados cambios. Reestructuración e Ingeniería Reversa caen dentro de la definición global de mantención de software.

2.2 CICLO DE VIDA Y ABSTRACCIONES

Para describir adecuadamente la noción de la ingeniería tradicional (Ingeniería Progresiva) e Ingeniería Reversa, primero se debe clarificar tres conceptos dependientes: la existencia de un modelo de ciclo de vida, la presencia de un sistema y la identificación de niveles de abstracción.

Se asume que existe un modelo de ciclo de vida para el proceso de desarrollo del sistema. El modelo puede ser representado como la tradicional Casada o Espiral.

El sistema sujeto en cuestión puede ser un programa singular o fragmentos de códigos o éste puede ser un complejo conjunto de programas interactuando y archivos de datos que en Ingeniería Reversa, es generalmente el punto de partida de este proceso.

Las fases del ciclo de vida tradicional envuelven una transición de alto nivel de abstracción a una representación de bajo nivel de abstracción en las sucesivas representaciones.

Este proceso sigue una secuencia que va desde la definición de los requerimientos, al diseño y de éste a la implementación.

2.3 REUSABILIDAD DE SOFTWARE

Por Reusabilidad de Software se entiende el repetido uso de cualquier parte estándar de un sistema: documentación, código, diseño requerimientos, casos de prueba, datos de prueba, etc..

Para ser verdaderamente estándar, un componente debe estar caracterizado por una especificación del tipo de **Caja Negra**. Los detalles de construcción del componente deben ser ocultos para el usuario y su especificación externa debe de ser suficiente para caracterizarla.

Para poder utilizarse, será necesario que se desarrollen y evolucionen familias de partes estándar. Y deberá ser posible que esas familias evolucionen y puedan descomponerse en subconjuntos de acuerdo con las demandas de uso.

El costo en el desarrollo de software está incrementando, mientras el costo del hardware está disminuyendo. Muchas de las razones para el decremento es la estandarización de partes comunes de hardware. Se puede ser capaz de lograr los mismos beneficios practicando el Reuso en Ingeniería de Software.

2.3.1 TIPOS DE REUSO

Existen dos formas de Reuso, cada una dependiendo sobre la perspectiva del reusador. **Reuso Productor** es la creación de componentes reusables y **Reuso Consumidor** es el uso de tales componentes en futuros sistemas.

Muchos de los avances en Ingeniería de Software hacen la técnica de Reuso más factible en su utilización. Dentro de estos avances está la adopción del Desarrollo Orientado al Objeto (DOO). Las herramientas Case que incluyen un depósito para los objetos y componentes desarrollados para un sistema pueden estimular el Reuso de estos objetos en posteriores sistemas.

Dos puntos de vista al Reuso son los de **Composición y Generativo**. El **Reuso de Composición** ve los componentes reusables como un conjunto de construcción de partes, y el desarrollo es hecho en forma Bottom-Up, es decir, la construcción del resto del sistema alrededor de componentes reusables disponibles. El **Reuso Generativo** es específico para un dominio particular de la aplicación, es decir, los componentes son diseñados específicamente para satisfacer las necesidades de una aplicación particular y para ser reusados posteriormente en una aplicación similar.

2.3.2 TECNOLOGIA DE REUSO Y RECUPERACION DE COMPONENTES

Uno de los más grandes obstáculos para el Reuso es el de encontrar a través de un gran conjunto de productos de software, el componente que satisfaga mejor la necesidad

particular del usuario. La solución a este problema es describir a cada componente por una lista ordenada de características llamadas Facetas. Una Faceta es una forma de descriptor que ayuda a identificar los componentes. Por ejemplo, las Facetas para el código reusable pueden ser las siguientes:

- Area de aplicación
- Función
- Lenguaje de programación
- Sistema operativo

Cada módulo de código (componente) es identificado por una entrada para cada una de las cuatro Facetas anteriormente mencionadas.

2.3.3 BENEFICIOS DE REUSO

El Reuso promete el incremento de la productividad y calidad durante todo el proceso de desarrollo de software. La productividad se está incrementando no solamente por la generación de documentación. Algunas de las más grandes reducciones de costo son ofrecidas por el reuso de requerimientos y diseño, éstas son áreas donde los errores pueden conducir a un costo extra.

2.4 REJUVENECIMIENTO DE SOFTWARE

El Rejuvenecimiento de Software está dirigido hacia el desafío de la mantención, por tratar de incrementar la calidad de un sistema existente. Existen muchas formas de Rejuvenecimiento de Software a considerar(figura 2.1), incluyendo:

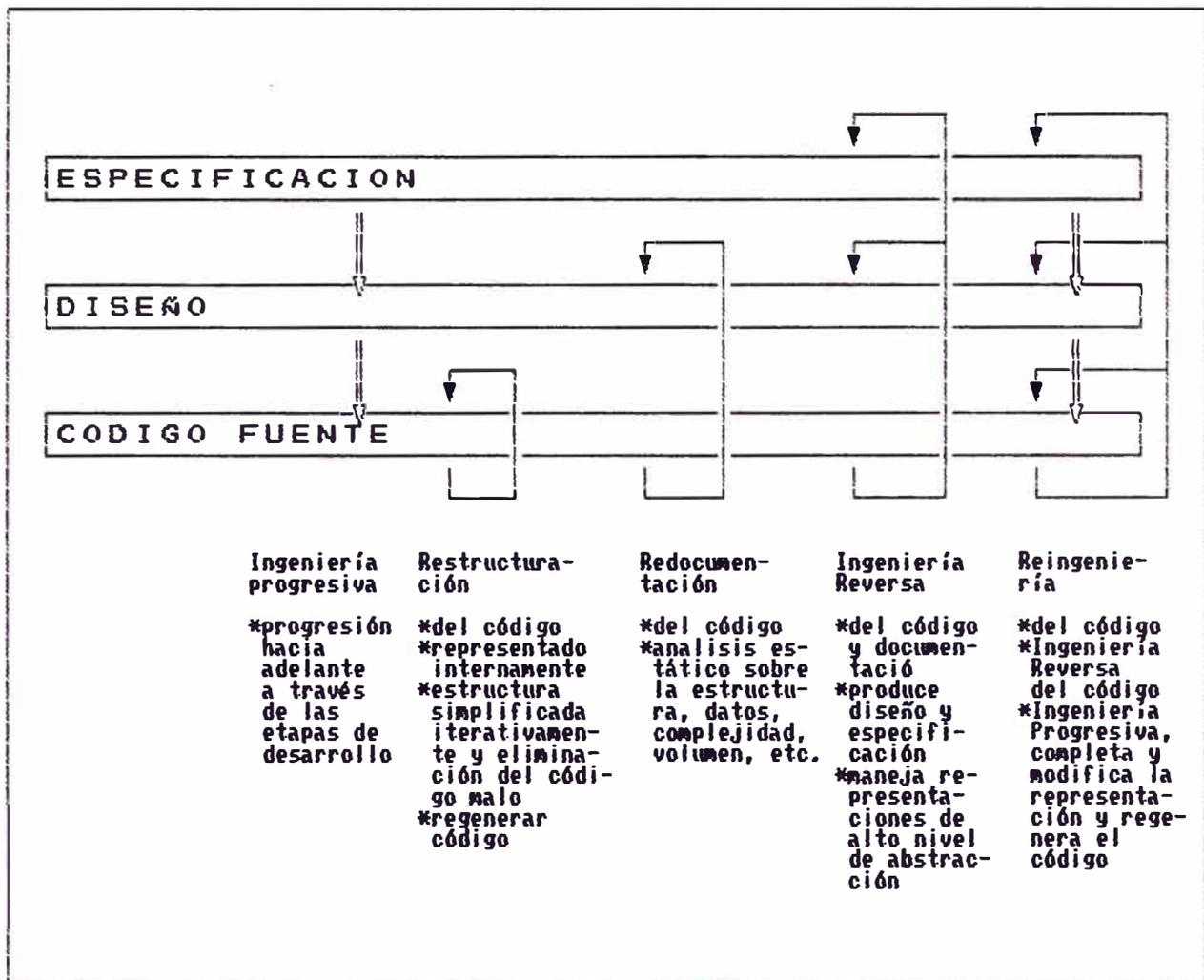


Figura 2.1 Comparación entre procesos relacionados

- Redocumentación
- Reestructuración
- Reingeniería
- Ingeniería Reversa

El grado al cual la información puede ser extraída del producto final depende sobre muchos factores:

- Lenguaje usado
- Interfases de Base de Datos
- Interfases de usuarios
- Interfases para servicios del sistema
- Interfases para otros lenguajes
- Herramientas disponibles

Además de los factores anteriormente mencionados, la habilidad, el conocimiento y la experiencia del mantenedor juega un rol importante en el grado que puede alcanzar la información.

2.4.1 REDOCUMENTACION

Redocumentación es la creación o revisión de una representación equivalente a un objeto dentro del sistema, a un mismo o distinto nivel; de abstracción ver figura 2.2.

Típicamente, un mantenedor comienza la Redocumentación para someter al código fuente a una herramienta de análisis estático.

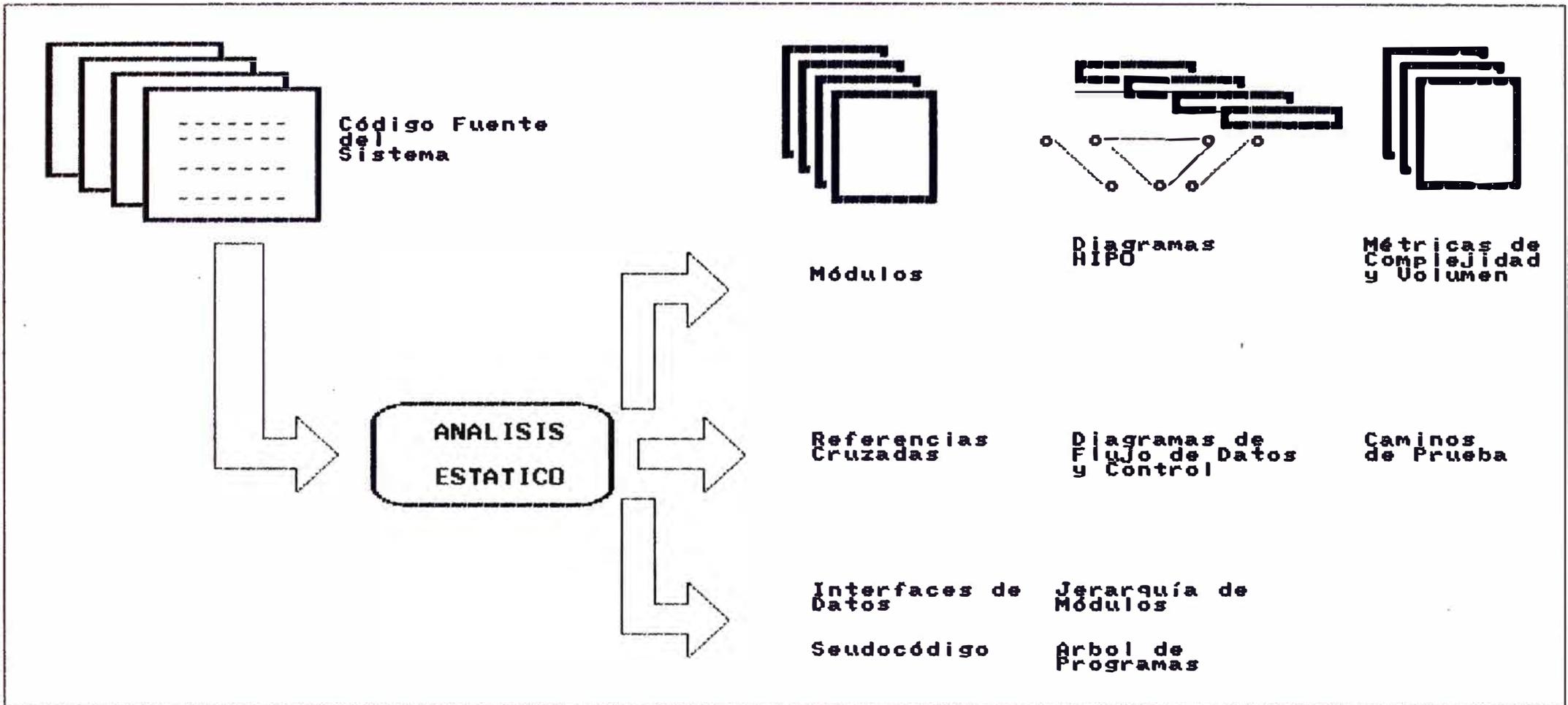


Figura 2.2 Proceso de Redocumentación

Las salidas pueden incluir:

- Diagramas Hipo

- Interfaz de datos

- Diccionarios de datos

- Flujos de datos

- Flujos de control

- Seudocódigos

- Referencias cruzadas de módulos y variables

2.4.1.1 RECUPERACION DE DISEÑO

La Recuperación de Diseño recrea la abstracción de diseño de una combinación de código, documentación del diseño existente (si está disponible) y además juega un papel muy importante la experiencia personal y el conocimiento general sobre el dominio o contexto del problema. Este proceso debe reproducir toda la información requerida por el analista para lograr un completo entendimiento de lo que hace el sistema, como éste lo hace y porque lo hace.

Así, la Recuperación de Diseño es común, algunas veces es la parte oculta de muchas actividades esparcidas por todo el ciclo de vida del software.

2.4.1.1.1 ETAPAS DEL PROCESO RECUPERACION DISEÑO

Un objetivo de la Recuperación de Diseño es desarrollar estructuras que ayudarán a la Ingeniería de Software a entender un sistema o programa. El entendimiento es importante para muchas actividades, tales como mantención, ampliación, Reuso, el diseño de un sistema nuevo similar.

El proceso de recuperación de Diseño consiste en dos pasos importantes:

Primera Etapa: El analista primero busca por una gran escala de estructuras organizacionales, tales como las estructuras de subsistemas, módulos y las importantes estructuras de datos (figura-2.3). Luego, recupera las estructuras de diseño y las representa en forma abstracta, tales como, diagramas de flujos de datos, diagramas de flujo de control, etc.. En el curso de ésta, la Ingeniería de Software mantiene el estudio de las relaciones entre las abstracciones y los segmentos de código que implantarán éstas.

¿ Qué son los módulos ?

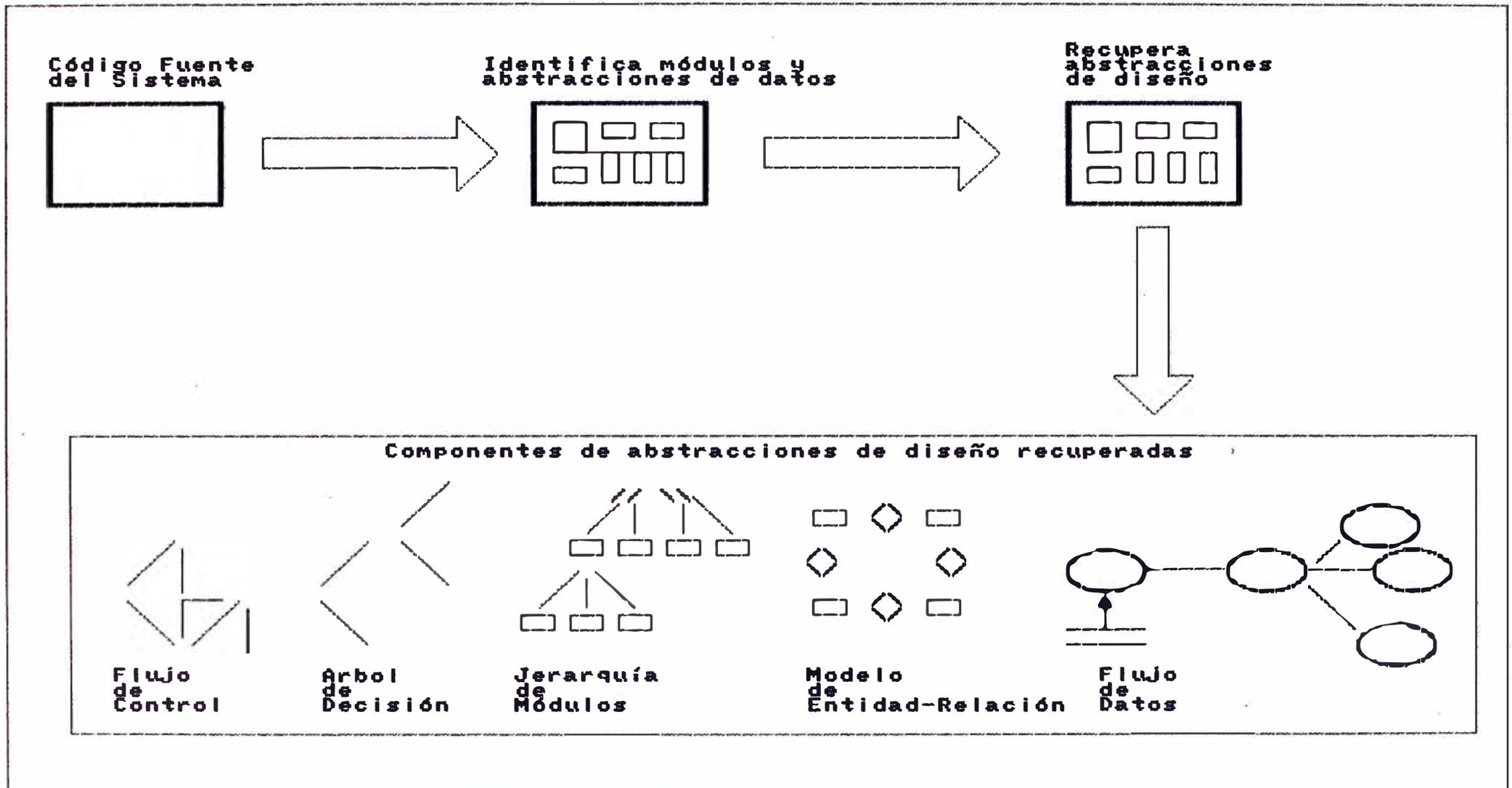
Son grupos de códigos que cumplen una determinada función; para identificar estructuras tales como módulos, se necesita también asociar las estructuras con conceptos semánticos informales. Esto es, se necesita proveer semánticamente una óptima abstracción del lenguaje natural o abstracción conceptual, que representa el concepto fundamental del módulo.

¿ Qué son los componentes de Ingeniería de Software ?

Como se muestra en la figura 2.3, el proceso de entendimiento recrea los componentes de diseño orientados a la Ingeniería de Software. Por ejemplo, lenguajes de descripción de programas o procesos, flujos de datos, diagramas jerárquicos, diccionario de datos.

Segunda Etapa: Apoyando la proliferación de Reuso.

¿ Cómo podemos usar productivamente los componentes recuperados ?



F13 Proceso de Recuperación de Diseño

Poblar la librería de componentes de Reuso de un sistema es beneficioso, pero éste requiere además procedimientos para generalizar los componentes con la finalidad de ampliar su reusabilidad. Las generalizaciones hacen a los componentes aplicables a un rango más amplio de las aplicaciones.

2.4.2 REESTRUCTURACION

Reestructuración es la transformación de una forma de representación a otra relativa, que por lo general es al mismo nivel de abstracción preservando la conducta externa del sistema (funcionalidad).

El término "Reestructuración" viene del popular uso de transformación código a código. Este rescata un programa de una forma no estructurada y lo transforma a una forma estructurada, también la normalización de datos, es una transformación de Reestructuración dato a dato para mejorar un modelo de datos lógico en el proceso de diseño de Base de Datos.

Reestructuración modifica el software para hacerlo más fácil de entender y mantener.

La figura 2.4 muestra las tres más importantes actividades de Reestructuración:

- 1.- Antiguamente la necesidad para Reestructuración era determinada por una herramienta de análisis estático u otros indicadores de mantención, el código fuente no estructurado es analizado y representado estructuralmente como una red semántica o grafo dirigido.

- 2.- La representación es refinada a través de sucesivas simplificaciones basadas sobre técnicas de transformación.

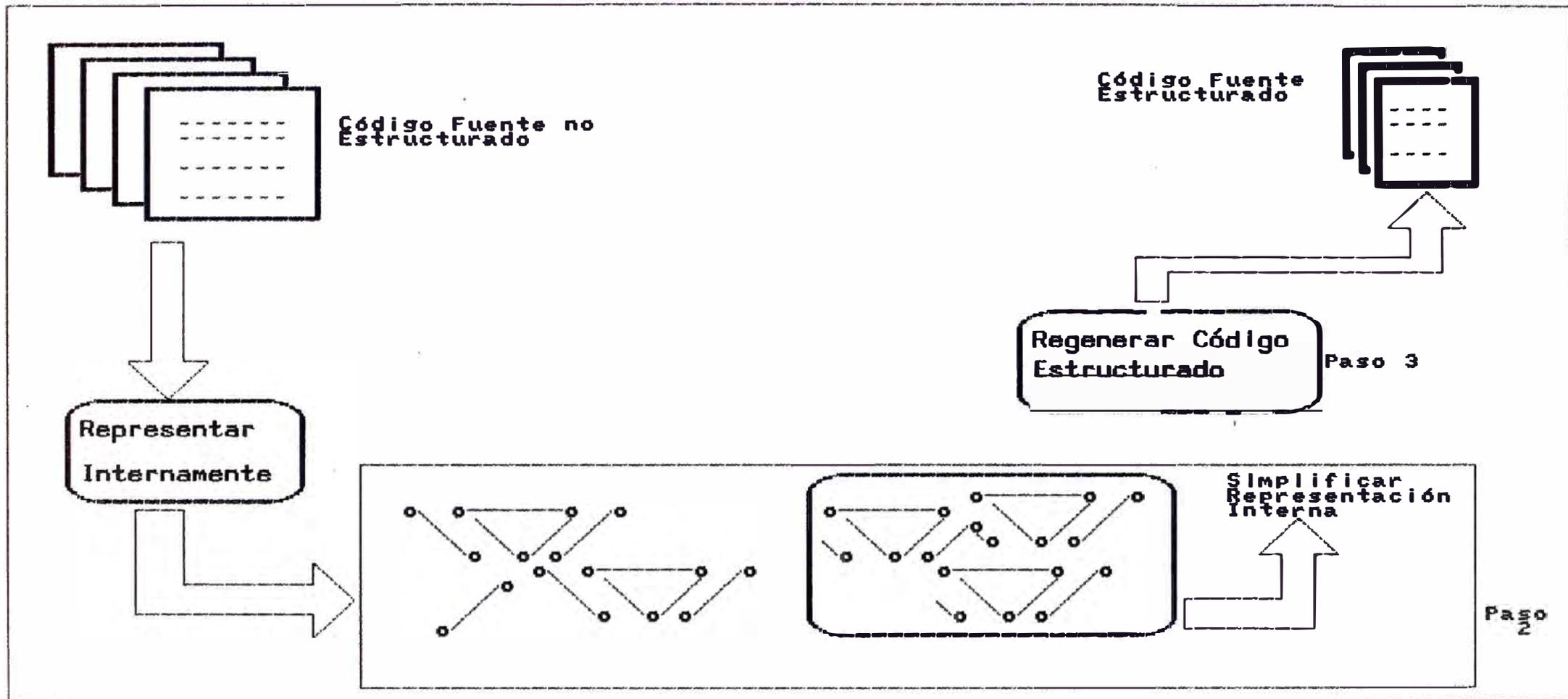


Figura 2.4 Proceso de Reestructuración

3.- La nueva representación es interpretada y devuelta como estructurada, pero equivalente al código anterior del sistema (típicamente para el mismo compilador), la cual fue lograda a través de una serie de transformaciones.

2.4.2.1 FORMAS DE REESTRUCTURACION

El siguiente es un método para reemplazar la sentencia "Goto" con la sentencias Case. La sentencia Case es ejecutada repetidamente en el interior de un ciclo Do-While, hasta que se cumpla la condición de salida del ciclo, indicando con esto el final de la función o módulo.

2.4.2.2 LA PROMESA DE REESTRUCTURACION

La Reestructuración intenta la complejidad por transformación de diseños de flujo de control estructurados con la finalidad de mejorar la legibilidad y la comprensión, mientras se mantiene la equivalencia funcional.

2.4.2.3 LOS PROBLEMAS DE REESTRUCTURACION

Los reestructuradores tienen un problema en que ellos no pueden identificar la estructura utilizando construcciones tales como Goto, por lo tanto las sentencias Case pueden ser simuladas

Comentarios de Programas

Otro problema de la Reestructuración es la pérdida de comentarios o los comentarios son consistentes con el correspondiente código fuente durante la Reestructuración. Los comentarios describen un bloque de código y son irrelevantes como código propiamente tal.

2.4.3 REINGENIERIA

Una simple definición de Reingeniería de Software es construir un nuevo sistema en base a uno existente. Otra, según Chikofsky and Crodd, "es la examinación y alteración del sistema de software para reconstruir éste en una nueva forma y la subsecuente implementación de la nueva forma".

Este podría ser contrastado con Ingeniería Reversa, la cuál "es el proceso de análisis del sistema para identificar los componentes del sistema y sus interrelaciones para luego crear una representación en otra forma o en una forma de abstracción de alto nivel". (figura 2.5).

Las entradas al proceso de Reingeniería incluyen a los archivos de código fuente, archivos de Base de Datos, archivos de generación de pantallas y archivos relativos al sistema similares a los anteriores. Cuando el proceso es completado éste genera toda la documentación del sistema, incluyendo especificaciones, diseño y el nuevo código fuente del sistema.

Existe alguna confusión de términos, entre Reingeniería y Reestructuración. Una aplicación de Reingeniería es también conocida como renovación y restauramiento, que es el proceso de: examinar y modificar el mecanismo interno de un sistema, de un programa o estructura de datos; sin cambiar su funcionalidad. En otras palabras, la Reingeniería está alterando el "como" sin afectar el "que".

Reingeniería incluye: Ingeniería Progresiva, Reestructuración e Ingeniería Reversa; y no por esto es un supertipo de ellas. Reingeniería utiliza estas tecnologías disponibles para hacer más eficiente y metódica su utilización.

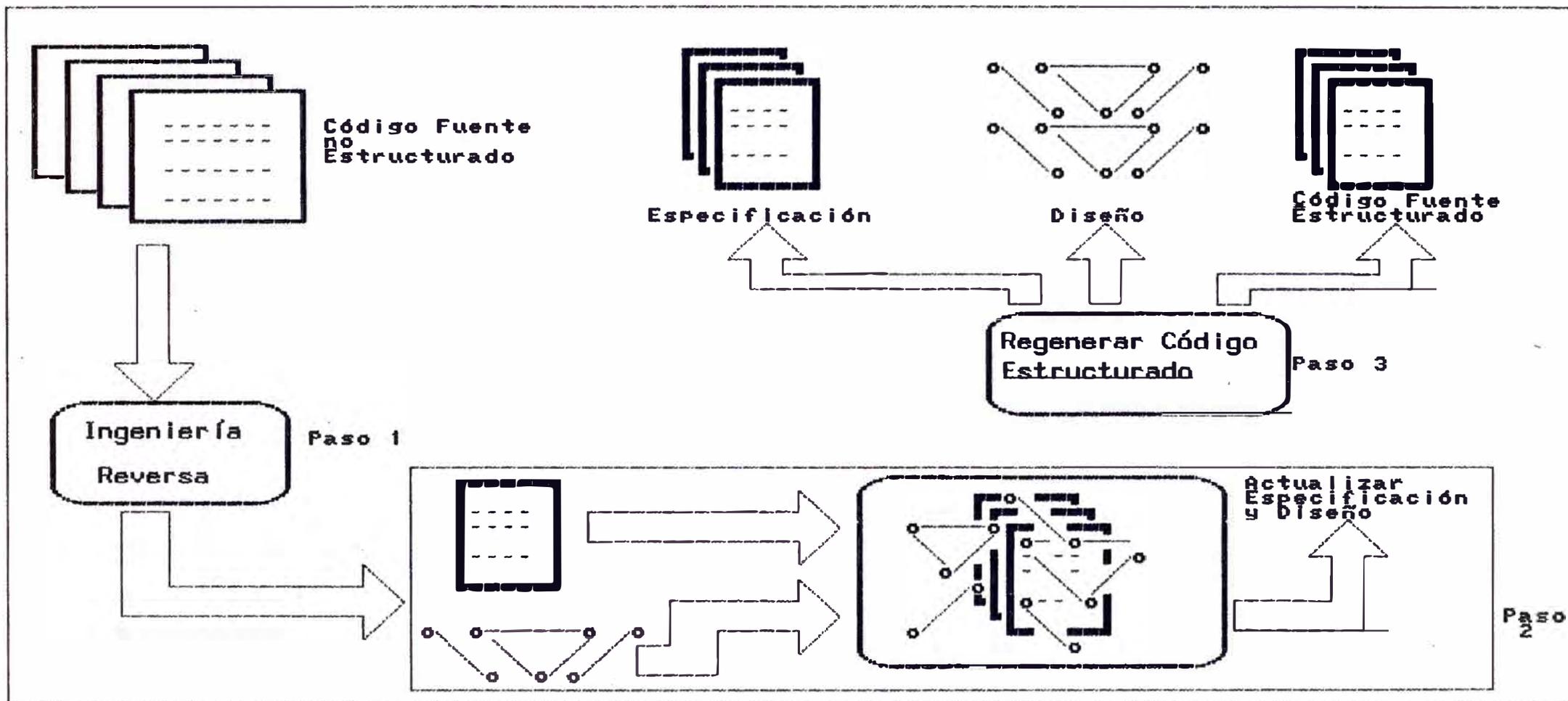


Figura 2.5 Proceso de Reingeniería

2.4.3.1 INTRODUCCION A LA TRANSFORMACION DE CODIGO FUENTE EN REINGENIERIA DE SOFTWARE

Reingeniería de Software es localizada primeramente sobre la transformación de la estructura del código fuente en alguna otra forma. Ingeniería Reversa está localizada sobre la transformación de código en información abstracta. Este tópico está abocado a la transformación de la estructura de código fuente en abstracciones, tales como especificaciones de diseño.

Reingeniería de Software es importante porque su sinergia con Ingeniería Reversa son la fuerza para las herramientas Case. Esto es porque no es óptimo reescribir el Software existente. El Software existente debe frecuentemente ser ampliado, ventaja de esta nueva tecnología, sin embargo, primero debe ser entendido, modificado y adaptado al nuevo ambiente. Reingeniería es una solución de compromiso impuesta por los altos costos de presupuestos y tiempo.

Hay muchas razones para utilizar Reingeniería sobre software existentes. Algunas razones son las siguientes:

- Migración de sistemas de un lenguaje a otro, un sistema de Base de Datos para un sistema operativo a otro, o de una plataforma hardware a otra. Obsolescencia de lenguajes, tecnología de acceso de datos, sistemas operativos.
- Carecer de confiabilidad y mantención en el software existente. El sistema puede tener una alta tasa de fallas. En esencia, esto es para reducir los costos de mantención.
- Reducción de la complejidad.
- Incremento en el desempeño del sistema.

Extensión de la vida útil del sistema.

2.4.3.2 COMPLEJIDAD Y COSTO DE MANTENCION

Una de las principales razones para la Reingeniería de software puede ser el factor del presupuesto de software que está siendo gastado en la mantención de software.

La Reingeniería hoy en día usa un modelo asumiendo que la complejidad del sistema es alta. Complejidad de los sistemas son de dos tipos: Complejidad Computacional y Complejidad Psicológica.

"Complejidad Computacional: los aspectos cuantitativos de soluciones para problemas computacionales, tales como la comparación de la eficiencia de las alternativas de soluciones algorítmicas".

"Complejidad Psicológica: Las características no anheladas de los programas computacionales que dificultan el entendimiento o el trabajo".

Las herramientas que intentan reducir la complejidad incluyen Reestructuración de Software, traducción de programas de software en otros y lenguajes estructurados.

2.4.3.3 REINGENIERIA DE PROCEDIMIENTOS

Reingeniería de procedimientos es un proceso que reestructura el flujo de control dentro de los procedimientos. Algunos tipos de Reingeniería, tales como Reestructuración de Programas, pueden ser completamente automatizadas. Otros pueden confiar en el discernimiento de los analistas.

2.4.3.3.1 REESTRUCTURACION DE PROGRAMAS

De acuerdo a Chikofsky and Cross, Reestructuración es "la transformación de una representación a otra al mismo nivel de

abstracción, mientras se preserve la conducta externa del sistema (funcionalidad y semántica).

2.4.3.3.2 ELIMINACION DE CODIGO REDUNDANTE

La redundancia es la implementación de funcionalidad duplicada en una aplicación, con las mismas características en repetidas partes del código, en múltiples módulos u ocurrencias los cuales se trata de eliminar. Porque la mantención de software, está llegando a tener un gran incremento en el porcentaje de presupuesto de software, código redundante significa más gastos tratando de mantener código duplicados o repetidas secciones de código.

2.4.3.3.3 ELIMINACION DE CODIGO MALO

El objetivo en eliminar el código malo es para reducir el gasto de tiempo de los programadores en entender el código o identificar dependencias asociadas con algunas cosas que tienen algún propósito o uso.

2.4.3.3.4 REMODULARIZACION

El objetivo de la modularización de software es para incrementar la mantenibilidad y disminuir el costo de desarrollo. Los módulos fuertemente cohesionados, integrados y ocultando información son el mejor criterio teórico para la modularización.

2.4.3.3.5 TRADUCCION DE LENGUAJES

Traducción es el proceso de convertir un programa de un lenguaje fuente a otro programa equivalente en un lenguaje destino. El objetivo primario de la traducción es mantener la funcionalidad, y el secundario es la eficiencia y la legibilidad (o mantención).

La nueva aproximación teórica incluye el uso de abstracción y reimplementación. Aproximadamente, los programas fuentes son primero analizados para ganar independientemente del lenguaje un entendimiento de la funcionalidad del programa. La fase de reimplementación genera un programa en el lenguaje destino.

2.4.3.4 REINGENIERIA DE DATOS

La Reingeniería de Datos intenta mejorar la mantención del sistema existente estandarizando la definición de datos.

Otros objetivos de la Reingeniería de Datos es mejorar el entendimiento del programa.

2.4.3.4.1 PROBLEMAS DE LA DEFINICION DE DATOS

Los típicos problemas de la definición de datos incluyen indescifrables o inconsistentes nombres o inflexibles longitudes de campos. Con esto se incrementa el esfuerzo requerido para los cambios de mantención. Los riesgos pueden también ser incurridos haciendo mejoras, alteraciones o descubrimientos de datos.

2.4.3.4.2 PROBLEMAS DE LOS VALORES DE DATOS

Los típicos problemas de valores de datos incluyen inconsistente valores por defecto (por ejemplo, espacios, ceros, etc.), la ambigüedad entre los valores por defecto y las pérdidas de valores, valores negativos en campos positivos, cambio de signos cuando el signo menos es omitido, overflow no detectado trunca los dígitos más significativos, inconsistentes unidades.

2.4.3.4.3 REINGENIERIA DE DATOS : EL PROCESO

La Reingeniería de Datos comienza con el análisis del código existente para descifrar la definición de datos, flujos

y reglas de validación. La información descubierta puede ser almacenada en un depósito de meta-dato (meta-dato es simplemente datos sobre datos).

Rediseño cambia la definición de datos en orden a mejorar el fundamental modelo de datos. Incluye pasos tales como racionalización de nombres de datos, estandarización de datos y reestructuración, las cuales son descritas abajo.

2.4.3.4.4 RACIONALIZACION DE NOMBRES DE DATOS

La racionalización de nombres de datos es el proceso de identificar los nombres de datos sinónimos a través de los programas y módulos y la eliminación de alias del código fuente.

La racionalización de nombres de datos puede ser un prerequisite para migración a un sistema DBMS (Database Management System).

2.4.3.4.5 REESTRUCTURACION DE DATOS

La Reestructuración de datos incluye cambios al modelo de datos físico, y como resultado, puede requerir un alto grado de cambio al código fuente, por el contrario, la estandarización y racionalización de datos solamente impactan al modelo de datos lógico.

2.4.3.4.6 NORMALIZACION DE DATOS

Primera Forma Normal

La Primera Forma Normal no permite atributos con multivalores dentro de una tabla. Cada atributo debe ser atómico, esto es simple e indivisible. En esencia esta forma no permite tablas sobre tablas.

Segunda Forma Normal

Una Tabla está en Segunda Forma Normal si cada atributo el cual no es miembro de la clave es completamente dependiente funcionalmente sobre la clave primaria para esta tabla. Si un atributo sólo depende sobre una parte de la clave primaria, la tabla puede ser descompuesta separando cada atributo el cual no es miembro de la clave primaria.

Tercera Forma Normal

Una tabla está en Tercera Forma Normal si cada atributo no-clave es completamente dependiente funcionalmente sobre cada clave de la tabla, y es dependiente no-transitivamente sobre cada clave de la tabla.

2.2.4 VISION FUTURA

Se están logrando importantes avances en la tecnología de Rejuvenecimiento de Software. Las herramientas comerciales de Ingeniería Reversa recuperan parcialmente el diseño de un sistema de software. Estas herramientas pueden identificar, presentar y analizar información del código fuente, pero ellos no reconstruyen, capturan y expresan la abstracción de diseño que no está explícitamente representada en el código fuente.

El código fuente no contiene mucha información sobre el diseño original, por lo tanto el resto debe ser reconstruido de inferencias.

El conocimiento informal sobre el dominio del problema y las aplicaciones son necesarias antes de un diseño completo para que éste sea entendido y reconstruido. Así el Rejuvenecimiento de Software avanzará cuando la tecnología y metodología puedan capturar reglas, políticas, terminologías, convenciones y otra información informal.

Al mismo tiempo, la formalización de las notaciones de diseño y la introducción de los modelos de dominio ampliarán la información disponible para el entendimiento y mantención de un sistema de software.

**CAPITULO III : APLICACION DE
INGENIERIA REVERSA
AL SISTEMA DE GESTION DOCENTE**

3.1 INTRODUCCION

La metodología de Ingeniería Reversa considera dos procesos principales: Redocumentación e Ingeniería Reversa propiamente dicha.

La aplicación de esta metodología se inicia con la etapa de Redocumentación del sistema existente; es decir, se captura la documentación, código fuente, módulos ejecutables y se definen los archivos tradicionales del sistema existente; y además se redocumentan aquellos procesos que carecen de documentación, todo esto con el objeto de realizar luego un análisis global de la calidad del sistema.

Recuperando la información del sistema existente se aplican secuencialmente las fases del proceso de Ingeniería Reversa para lograr con esto un suficiente entendimiento del sistema en cuestión y una documentación actualizada de éste.

Al finalizar se contará con una confiable documentación del sistema existente y un óptimo análisis de reusabilidad de los componentes identificados (recuperados) del sistema en estudio, para su posible futura utilización en el nuevo

sistema que se reimplementará, el cual tendrá una equivalente funcionalidad con el sistema original (existente).

3.2 OBJETIVOS

Primero se debe tener claro que en la aplicación el objetivo no es la metodología de Ingeniería Reversa, ya que ésta es sólo un método para lograr las metas fijadas. A continuación se definirán los objetivos perseguidos por esta aplicación.

Objetivo General

- Estandarizar los Sistemas de Información Administrados por el Centro de Cómputo de la Universidad Andina Néstor Cáceres Velásquez a la herramienta de desarrollo de sistemas Ingres, basados en el modelo de base de datos Relacional.

Objetivos Específicos

- Migrar el Sistema de Gestión Docente a la herramienta de desarrollo de sistemas Ingres.
- Aprovechar los recursos computacionales de la máquina Digital MicroVax 3100.
- Disminuir la alta congestión de la máquina S/34 de IBM.

3.3 DESCRIPCION DEL SISTEMA DE GESTION DOCENTE

3.3.1 REGIMEN ACADEMICO

3.3.1.1 ESTRUCTURA ACADEMICA

La Universidad Andina Néstor Cáceres Velásquez, se estructura académicamente por Facultades, estando integrados por Carrera Académico Profesionales, Departamentos Académicos, Institutos de Investigación, Escuela de Post Grado; que gozan de autonomía de gobierno, académica, económica y administrativa.

Las Facultades se organizan a través de: Secretarías Académicas y Administrativas que dependen del Decano.

Los Departamentos Académicos son: unidades Académicas de Servicio, integrados a una Facultad; asignan los docentes a las Facultades, determinan y actualizan los syllabi, de acuerdo con los requerimientos curriculares de las Facultades y necesidades de sus Carreras Académico Profesionales.

La UANCV tiene las siguientes Facultades y Carreras Académico Profesionales:

Facultad de Ciencias de la Salud:

- Carrera Académico Profesional de Obstetricia
- Carrera Académico Profesional de Enfermería
- Carrera Académico Profesional de Medicina Veterinaria y Zootecnia

Facultad de Ciencias Contables y Administrativas:

- Carrera Académico Profesional de Administración de Empresas
- Carrera Académico Profesional de Contabilidad

Facultad de Ingenierías y Ciencias Puras:

- Carrera Académico Profesional de Ingeniería Industrial
- Carrera Académico Profesional de Ingeniería Mecánica Eléctrica

- Carrera Académico Profesional de Ingeniería Civil
- Carrera Académico Profesional de Ingeniería de Sistemas

Facultad de Ciencias Jurídicas y Políticas:

- Carrera Académico Profesional de Derecho

Facultad de Ciencias de la Educación:

- Carrera Académico Profesional de Educación

Existen dos Departamentos Académicos:

Departamento de Ciencias Puras:

- Matemáticas, Física, Química y Estadística, integrado a la Facultad de Ingenierías.

Departamento de Letras:

- Sociología, Arqueología, Antropología, Psicología y Filosofía; integrado a la Facultad de Educación.

Cada Facultad establece el horario de clases de acuerdo a sus propios requerimientos y la capacidad instalada de la Universidad, en coordinación con el Vice-Rectorado Académico, en los turnos de mañana, tarde y noche.

3.3.1.2 DOCENTES UNIVERSITARIOS

Los Profesores de la UANCV, son aquellos que ejercen la docencia universitaria en su más alta expresión; en la ENSEÑANZA APRENDIZAJE, INVESTIGACION, PROYECCION SOCIAL, CAPACITACION PERMANENTE Y PRODUCCION INTELECTUAL de bienes y servicios, de acuerdo a los objetivos y fines que establece su Estatuto.

Para el ejercicio de la docencia universitaria en la Universidad Andina Néstor Cáceres Velásquez de Juliaca, es bligatorio poseer Grado Académico de Maestro o Doctor y/o

Título Profesional u otros conferidos por las Universidades del país o revalidados según ley.

El ingreso a la carrera docente, en condición de profesor ordinario, se hace por Concurso Público de méritos o prueba de capacidad docente y/o por oposición, de sus funciones con evaluación, calificación de acuerdo a los reglamentos respectivos.

Los profesores Universitarios son:

- a) Ordinarios
- b) Extraordinarios, y
- c) Contratados

3.3.1.2.1 DOCENTES ORDINARIOS

Los profesores ordinarios son admitidos por Concurso Público de méritos o por oposición, cuyas categorías son las siguientes:

- a) Profesores Principales
- b) Profesores Asociados
- c) Profesores Auxiliares

Profesores Principales.- Es la más alta categoría que puede lograr un docente universitario.

Profesores Asociados.- Es el grado inmediato al que es promovido un docente auxiliar.

Profesores Auxiliares.- Es la categoría que recibe un profesional cuando recién ingresa a la docencia.

Los profesores ordinarios de la UANCV, según su dedicación al trabajo universitario, pueden ser:

- a) A Dedicación Exclusiva
- b) A Tiempo Completo y
- c) A Tiempo Parcial

Profesor a Dedicación Exclusiva

Es aquel que dedica su tiempo y actividad a la docencia Universitaria como única actividad remunerada y que no presten servicios en otras instituciones cualquiera que fuere su denominación.

Profesor a Tiempo Completo:

Es aquel que labora y dedica su tiempo de cuarenta (40) horas semanales a las actividades académicas o administrativas de la UANCV.

Profesor a Tiempo Parcial:

Es aquel que dedica su labor académica en la Universidad, hasta veinte (20) horas semanales, según su antigüedad en la docencia.

El cambio de categoría o régimen se efectúa por evaluación de acuerdo a las posibilidades y necesidades de la Universidad y de conformidad al Reglamento pertinente.

El nombramiento de los profesores ordinarios en conformidad a la Ley Universitaria, tendrá la siguiente duración:

- a) Siete (7) años para los Profesores Principales
- b) Cinco (5) años para los Profesores Asociados
- c) Tres (3) años para los Profesores Auxiliares

3.3.1.2.2 REQUISITOS PARA CADA CATEGORIA DE PROFESORES ORDINARIOS:

Para ser **Profesor Principal** se requiere:

- a) Grado Académico de Maestro o Doctor .
- b) Haber desempeñado como mínimo cinco (5) años de docencia universitaria en la categoría de Profesor Asociado de la UANCV .
- c) Haber realizado trabajos de investigación, en el campo de su especialidad.

- d) Haber sido ratificado en la categoría de Profesor Asociado.
- e) Por excepción podrán concursar también a esta categoría, profesionales con reconocida labor de investigación científica y con más de diez (10) años de ejercicio profesional.

Para ser **Profesor Asociado** son requisitos:

- a) Tener título profesional universitario o grado de maestro o doctor.
- b) Haber desempeñado tres (3) años de docencia en la categoría de profesor auxiliar en la UANCV.
- c) Haber realizado trabajos de investigación en su especialidad.
- d) Haber sido ratificado en la categoría de Profesor Auxiliar.

Para ser **Profesor Auxiliar** son requisitos:

- a) Tener título profesional universitario o grado de maestro o doctor.
- b) Poseer tres (3) años de experiencia en la docencia universitaria y/o cinco (5) años de experiencia como mínimo en el ejercicio profesional.
- c) Haber realizado trabajos de investigación en su especialidad.

3.3.1.2.3 DOCENTES EXTRAORDINARIOS

Los profesores extraordinarios, son las personas de excepcional experiencia en un ámbito profesional, de una reconocida trayectoria en la enseñanza, investigación, proyección social y la producción científica, cultural y tecnológica. Pueden ser:

- a) Eméritos
- b) Honorarios

- c) Investigadores
- d) Visitantes

Los profesores EMERITOS:

Son aquellos profesores principales, jubilados o cesantes de la UANCV de reconocida labor científica y académica, designados por el Consejo Universitario, su nombramiento tendrá carácter vitalicio.

Los profesores HONORARIOS:

Son profesores nacionales o extranjeros, que sin tener carrera docente en la UANCV, se hacen merecedores a esta distinción por sus relevantes méritos académicos, reconocida producción científica, tecnológica y/o cultural. Esta distinción es conferida por el Consejo Universitario.

Los profesores INVESTIGADORES:

Son los que se dedican exclusivamente a la creación y producción intelectual. Son designados en razón de su excelencia académica, están sujetos a régimen especial que la UANCV determina en cada caso.

Los profesores VISITANTES:

Son profesionales y especialistas de otras instituciones nacionales o extranjeras que por cualquier forma de su colaboración aceptada por la Universidad, presta sus servicios. Son nombrados por el Consejo Universitario.

3.3.1.2.4 DOCENTES CONTRATADOS

Los profesores contratados, son los que prestan servicios a la Universidad, de acuerdo a las necesidades de cada Facultad y/o Carrera Académico Profesional.

Los profesores contratados los son por el plazo máximo de seis (6) años al término de éste, tiene derecho a concurso de

acuerdo a las necesidades de la UANCV. Al no efectuarse, el concurso puede o no renovarse por una sola vez y por el mismo plazo máximo, previa evaluación.

La contratación de docentes se realiza mediante una acción de evaluación conducido por una Comisión.

La Comisión para la contratación de Docentes está conformada por el Decano de la Facultad, Director de la C.A.P., Jefe de Departamento, Presidente de la Comisión Académico Curricular y el Docente de mayor jerarquía de la especialidad correspondiente.

Una vez realizada la evaluación respectiva la Comisión propondrá la contratación al Consejo de Facultad en una primera instancia y luego se procederá a proponer al Consejo Universitario en última instancia recomendado la emisión de una Resolución Rectoral y el contrato en la oficina de Personal.

Para ser contratado se requiere:

- a) Poseer Grado Académico de Maestro y/o Doctor y/o Título Profesional de nivel universitario.
- b) No tener mas de 45 años de edad.
- c) Poseer tres (3) años de experiencia en la docencia universitaria y/o cinco (5) años de experiencia como mínimo en el ejercicio profesional, con excepción para las C.A.P. de Ingenierías.
- d) Otros que determine la autoridad.

3.3.1.2.5 JEFES DE PRACTICAS

Son docentes universitarios sin categoría, los jefes de prácticas; cuyo nombramiento y/o contrato se hará en concordancia con el Reglamento de concurso de plazas docentes de la Universidad.

Para ser jefe de práctica, se requiere:

- a) Título Profesional o Grado Académico de Bachiller.
- b) Otros que señale el Reglamento de concurso.

Los jefes de prácticas, serán nombrados o contratados con un mínimo de tres (3) años de servicio. El tiempo de trabajo se computa con el tiempo de servicio en la docencia, para que obtenga la categoría de profesor auxiliar desde la fecha que ha obtenido su Título Profesional.

3.3.1.3 REGLAMENTO GENERAL DE DOCENTES

3.3.1.3.1 EVALUACION, PROMOCION Y RATIFICACION DE DOCENTES

La evaluación a la docencia universitaria tiene por objeto calificar el cumplimiento de las funciones inherentes del docente y que están enmarcados en los fines de Facultades, y/o Carreras Académico Profesionales y la Universidad.

Son fines de la evaluación del docente universitario los siguientes:

- a) Proporcionar los elementos de juicio para la ratificación y ascensos del profesor evaluado.
- b) Garantizar el cumplimiento y la eficiencia de las actividades docentes.
- c) Evaluar la labor del profesor de la UANCV en las áreas de docencia, investigación, proyección social y responsabilidad administrativa.

La ratificación de los docentes, es el resultado de la evaluación permanente y acumulativa, la misma que se ajusta a lo establecido en la Ley Universitaria y el Estatuto Universitario.

3.3.1.3.2 DEBERES Y DERECHOS DE LOS DOCENTES

La participación de los profesores en la vida de la comunidad universitaria tiene su más alta expresión en la docencia e investigación; consiste además, en su derecho a integrar los órganos de gobierno y las comisiones universitarias para las que sean elegidos o designados, así como intervenir en las actividades de proyección social y de extensión universitaria.

Los profesores en armonía a la Constitución Política del Estado la ley Universitaria, el Estatuto de la UANCV y demás normas vigentes, tiene los siguientes deberes:

- a) Ejercer la docencia y la investigación con libertad de pensamiento y con respeto a la discrepancia, dentro del marco de los fines, principios, objetivos y valores que inspiran a la UANCV.
- b) Perfeccionar permanentemente sus conocimientos y su capacidad docente y realizar labor intelectual creativa.
- c) Ejercer sus funciones en la Universidad con independencia de toda actividad política partidaria.
- d) Asesorar a los alumnos en trabajos de investigación, proyección social y extensión universitario, en forma ad-honorem.
- e) Participar como jurado dictaminador en los proyectos de investigación y/o tesis en los Grados y Títulos que la UANCV confiere.
- f) Cumplir con el dictado de clases de 45 minutos como mínimo de la hora pedagógica establecida.
- g) Hacer entrega de las pruebas calificadas y dar el informe a los estudiantes sobre las notas de evaluación, sean éstas escritas, orales, sustentaciones, trabajos encargados, exposiciones u otros dentro de los plazos establecidos.

Los docentes con régimen de dedicación exclusiva y tiempo completo tienen a su vez los siguientes deberes:

- a) Al inicio del semestre académico formular su respectivo plan de trabajo para su aprobación por el Consejo de Facultad e informadas al Vice-Rectorado Académico para fines de evaluación.

Los profesores ordinarios, en armonía con la Constitución Política del Estado, la Ley Universitaria y el Estatuto de la UANCV, tienen los siguientes derechos:

- a) El ejercicio de la docencia con libertad de pensamiento y respeto a la discrepancia de los estudiantes.
- b) Dedicar semestres al estudio y la investigación sin carga lectiva, de acuerdo a las normas establecidas para este fin.
- c) Postular a becas de perfeccionamiento y solicitar licencias, dentro de las normas establecidas para tal fin consideradas en el presente reglamento y el de becas para los profesores de la UANCV.
- d) Gozar del año sabático con fines de investigación o preparación de publicaciones de conformidad al respectivo reglamento.
- e) La publicación de su producción intelectual por parte de la UANCV, garantizando los derechos de autor, previa revisión y aprobación por el Consejo Universitario.

3.3.1.3.3 SANCIONES

Las sanciones son medidas disciplinarias excepcionalmente aplicables en casos comprobados de incumplimiento de los deberes de los profesores y por atentar contra los derechos de los docentes, estudiantes, trabajadores no docentes o del patrimonio de la UANCV.

Las sanciones por aplicarse a los docentes son:

- a) Amonestación privada
- b) Amonestación pública
- c) Multa
- d) Suspensión temporal
- e) Separación definitiva.

3.3.1.3.4 INCOMPATIBILIDADES Y PERDIDA DE LA DEDICACION EXCLUSIVA Y TIEMPO COMPLETO.

Sólo los docentes ordinarios de la UANCV están en la capacidad de ejercer regímenes de dedicación.

En cualquier momento el profesor de la UANCV, puede renunciar a la Dedicación Exclusiva y/o Tiempo Completo; o ser privado de ésta, si se comprueba que ha incurrido en las causales que ocasionan su pérdida de régimen de dedicación y devolverá los haberes indebidamente cobrados, sin perjuicio de ser sometidos al proceso pertinente.

3.3.2 CARGA ACADEMICA DOCENTE

Las Jefaturas de Departamento Académico son los encargados de asignar la distribución de la Carga Académica a los Docentes a solicitud de las Decanaturas de las diferentes Facultades y/o Direcciones de las Carreras Académico Profesionales (C.A.P.). La carga académica es aprobada por el Consejo de Facultad.

3.3.2.1 CRITERIOS DE ASIGNACION DE CARGA ACADEMICA:

Para la distribución de la Carga Académica se deberá tomar en cuenta los siguientes criterios:

1. Categoría del Docente.

2. Régimen de Dedicación

3. Antigüedad en la Universidad

4. Profesión y/o Especialidad afín a la asignatura a dictarse.

3.3.2.2 CARGA LECTIVA

La carga lectiva para los profesores con régimen de Dedicación Exclusiva será no menor de veinte (20) horas semanales.

La carga lectiva para los profesores con régimen de Dedicación a Tiempo Completo será de veinte (20) horas semanales.

La carga lectiva para los profesores con régimen de Dedicación a Tiempo Parcial será de conformidad a su nombramiento.

En asignaturas de Formación Especializada, el Docente tendrá la responsabilidad de desarrollar un máximo de cuatro (4) asignaturas.

La carga lectiva para docentes con cargo administrativo será como sigue:

1. Rector y Vicerrectores: cinco (5) horas como máximo.
2. Decanos : ocho (8) horas como mínimo y diez (10) horas como máximo.
3. Directores de C.A.P. : quince (15) horas como máximo.
4. Jefes de Oficina : quince (15) horas como máximo.

El Decano asigna la carga académica mediante memorándum en primera instancia y es ratificada por Resolución Rectoral en última instancia.

Recepcionada la carga académica, el Docente elabora en original y tres copias el Syllabi y entrega a su Facultad y/o C.A.P. en un plazo máximo de diez (10) días hábiles de recepcionado el memorándum. La distribución de los syllabi es como sigue: Una para la Oficina de Servicios Académicos, una para el archivo de la Facultad y/o C.A.P., una para el docente y una última para los estudiantes.

3.4 ESTUDIO PRELIMINAR DEL PROBLEMA

Como se sabe la metodología de Ingeniería Reversa no tiene el tradicional sentido de las otras metodologías de análisis de sistemas, estas van siempre desde las especificaciones al diseño, en cambio Ingeniería Reversa es un proceso inverso que va desde el código fuente a las especificaciones. Esto es bastante útil cuando se cuenta con sistemas en funcionamiento, a los cuales se les desea realizar un proceso de mantenimiento y/o migración a otros ambientes de desarrollo u operación y además si estos sistemas cuentan con una regular documentación. Condiciones que se cumplen para el caso en aplicación.

El sistema objeto que se desea "reversar" (aplicar la metodología de Ingeniería Reversa al sistema en cuestión) es el sistema de Gestión Docente existente en la universidad Andina de Juliaca por requerimientos de cambios en la infraestructura de software y hardware, necesitando por lo tanto una migración o conversión.

Esta metodología permite obtener y/o recuperar del Sistema de Gestión Docente (sistemas existentes) abstracciones de alto nivel, es decir, diccionarios de datos, diagramas de entidad-relación, árbol de menús y diagramas del Análisis y

Diseño Estructurado (diagrama de contexto, flujos de datos, descripción de procesos, etc.), las cuales son representaciones del sistema (documentación) que proporcionarán la información necesaria para lograr un acabado entendimiento del sistema existente. Toda esta información o documentación será suficiente para regenerar el sistema en el nuevo ambiente de desarrollo **Ingres** (Ver Anexo No 2).

El sistema ha *reversar* está desarrollado bajo una filosofía de lenguaje de tercera generación Cobol, con archivos tradicionales. Este se desea convertir o migrar a un ambiente desarrollador de sistemas de información basados en el modelo de Bases de Datos relacional Ingres, el cual utiliza un lenguaje de cuarta generación. Todo esto con la finalidad de contar con una plataforma base que permita la eficiente implementación de futuras mantenciones. Al igual que en el soporte software, el soporte hardware también necesita ser cambiado desde una máquina IBM S/34 a la máquina MicroVax 3100 dada la necesidad de disminuir la congestión existente en la máquina original y por limitantes en los recursos de memoria.

La entidad organizacional dentro de la Universidad de Juliaca que está encargada del desarrollo, mantención y operación del Sistema de Información de Gestión Docente es el Centro de Cómputo.

El sistema existente está destinado a entregar información relativa a las distintas actividades docentes semestrales, que se realizan en la Universidad de Juliaca a los diferentes niveles organizacionales que las requieran.

El Sistema de Gestión Docente debe registrar y mantener la información correspondiente a:

- Grupos de Asignaturas
- Cargas Docentes
- Parámetros docentes

- Facultades
- Departamentos
- Carreras
- y otras informaciones relativas a las actividades docentes.

También este sistema debe interrelacionarse con otros sistema que administren información correspondiente a:

- Curriculas de estudios
- Inscripciones de asignaturas
- Asignaturas
- Funcionarios de esta casa de estudios

Entre las salidas que debe emitir el sistema se tienen las siguientes:

- Solicitudes docentes por departamento
- Solicitudes docentes por especialidad
- Cargas docentes por departamentos
- Cargas docentes por profesor
- Curriculas vigentes
- Secciones por departamentos
- y otras salidas requeridas por los usuarios

Entre los procesos que debe considerar el sistema se cuentan los siguientes:

- Solicitudes de asignaturas
- Conformación de grupos de asignaciones
- Asignación de profesores a los grupos
- Emisión de informes
- Responder consultas
- y otros procesos relativos a las actividades docentes.

Este sistema originalmente se abordó en su etapa de análisis y diseño con la metodología de Análisis y Diseño Estructurado.

Todas las etapas involucradas en Ingeniería Reversa son realizadas manualmente, porque no se dispone de herramientas automatizadas a nuestro alcance que apoyen las tareas que involucra esta metodología.

En la ilustración siguiente (figura 3.1) se muestran los procesos utilizados durante el análisis de Ingeniería Reversa y sus respectivas entradas y salidas. Además se muestra la fase de Reimplementación de Sistema, la cual no es parte de la metodología de Ingeniería Reversa y no se incluye en detalle en este trabajo.

La primera etapa cubre el proceso de Redocumentación y Recuperación de Diseño, la cual obtiene y/o captura la información o documentación correspondiente al sistema existente dejándola disponible para que sirva como entrada al proceso de Ingeniería Reversa. Las entradas a este último proceso son las siguientes: definición de archivos Cobol, cierta documentación del sistema, archivos de datos, códigos fuentes y códigos ejecutables del sistema existente. A partir de estas entradas Ingeniería Reversa generará salidas que serán ubicadas teóricamente en un depósito (ver Anexo No 3). Luego esta información depositada, sirve como entrada a la fase de Reimplementación, es decir, capturamos las especificaciones y el diseño para regenerar el sistema reflejado en la documentación, que será de igual funcionalidad y mejor calidad. La última fase de Reimplementación será desarrollada bajo la herramienta Ingres, la cual generará como salidas un nuevo código fuente, código ejecutable y Base de Datos.

A continuación se desarrollaran los procesos involucrados en la metodología de Ingeniería Reversa.

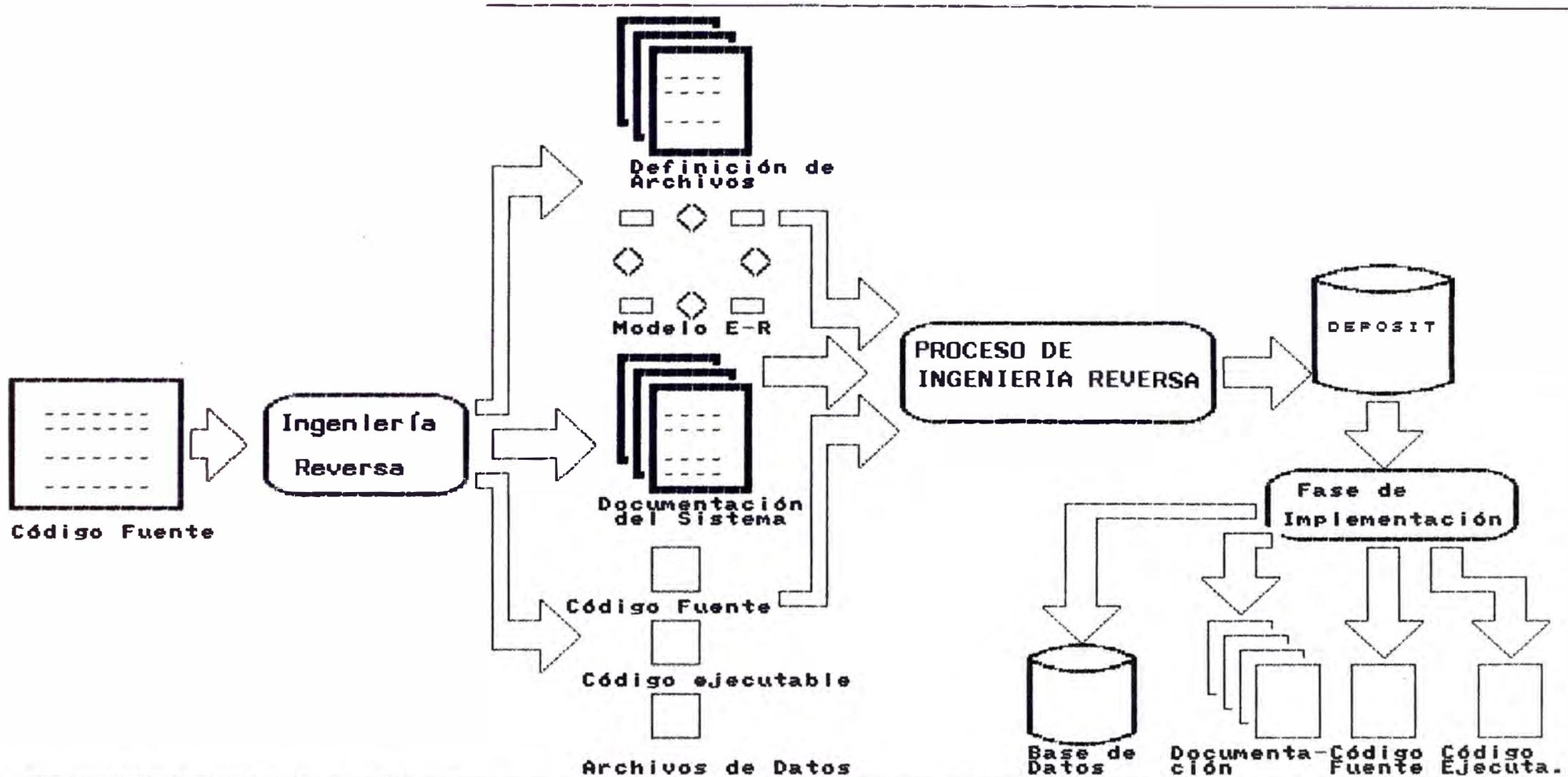


Figura 3.1 Entrada/Salida a los procesos involucrados en la aplicación de la metodología

3.5 PROCESO DE REDOCUMENTACION

Esta etapa tiene por objetivo obtener (capturar, generar o regenerar) la documentación correspondiente al sistema de Gestión Docente existente. Esta documentación puede existir actualmente o pudo haber existido hace tiempo atrás o quizás nunca existió. De acuerdo a estas alternativas será necesario capturar, regenerar o generar respectivamente estas representaciones del sistema existente.

De acuerdo a las entradas de este proceso, es decir, códigos fuentes, como lo muestra la figura 3.1, se capturan las especificaciones, el diseño, y la documentación o cualquier información relativa que esté dentro del dominio del problema o aplicación. También se deben capturar las interrelaciones que existen entre estos objetos capturados, como por ejemplo, cada archivo ejecutable está relacionado con un archivo fuente y además toda esta información, es decir, especificaciones, diseño y relaciones deben estar reflejadas en un archivo de documentación que deberían estar en el sistema existente. Cuando no ocurre el caso anterior es necesario generar o regenerar documentación para estos objetos. Luego de haber obtenido esta información se está en condiciones de evaluar la información que se mantiene del sistema con respecto a su calidad.

Después del proceso de redocumentación se tendrá disponible las entradas al proceso principal de Ingeniería Reversa, es decir, definiciones de archivos, archivos de datos, la documentación existente del sistema, códigos fuentes y códigos ejecutables sobre los cuales se realizará el análisis de reusabilidad.

3.6 PROCESO DE INGENIERIA REVERSA

La etapa anterior dejó disponibles las entradas para este

proceso, a partir de las cuales se obtendrán como salida una representación confiable del sistema Gestión Docente.

A continuación, se van a desarrollar las etapas involucradas en el proceso de Ingeniería Reversa para con esto lograr un suficiente entendimiento, documentación confiable del sistema en estudio y un óptimo análisis de Reusabilidad de los componentes.

3.6.1 Etapa de Planificación

Como un primer paso dentro de esta fase, se realiza un análisis de calidad del sistema existente con respecto a la consistencia entre la documentación y la implementación del sistema, es decir, recopilados todos los procesos (programas) y definiciones de archivos que son parte del sistema existente se verifica la consistencia. Se realiza a cada proceso un análisis de Caja Negra, es decir, un análisis de entrada y salida y se compara con la descripción del proceso correspondiente. También se verifica la consistencia de las definiciones de archivos Cobol con sus respectivas descripciones de archivos correspondientes a la documentación existente. Además se verifica la consistencia a nivel de jerarquía de módulos con el documento de árbol de menús. Todo esto realizado por la posibilidad de que se hayan realizado mantenciones y no se hayan reflejado en la documentación y para medir el nivel de calidad de la documentación existente.

De acuerdo al análisis global de calidad realizado al sistema de Gestión Docente existente, se obtuvo como resultado un regular nivel de aceptación de la documentación, es decir; la documentación existente es recuperable para su utilización en la fase de reimplementación o para futuras mantenciones. Después de este análisis general de la calidad del sistema con respecto a la confiabilidad de la documentación existente, se inicia la primera actividad dentro de esta etapa de planificación.

3.6.1.1 Analizar los Objetivos del Proyecto

El Objetivo clave en este proyecto es migrar o convertir el sistema de Gestión Docente existente en la Universidad de Juliaca a la herramienta Ingres. El soporte software del sistema existente es el lenguaje de programación Cobol bajo el sistema operativo OS/34, utilizando herramientas de apoyo a la administración de pantallas, menús, informes; como los siguientes SDA (generador de pantallas apoyados por ventanas) SAM (Sistema Administrador de Menús) y RLU (Generador de Reportes). El soporte hardware es la máquina IBM S/34.

El nuevo ambiente al cual se desea migrar este sistema cambia completamente la infraestructura software y hardware por lo que será necesario recodificar en un ciento por ciento el sistema en estudio. El nuevo ambiente donde se desea desarrollar el sistema tiene el siguiente soporte software:

- Herramienta Ingres V6.0
- Sistema Operativo Ultrix V4.2

El soporte hardware es la máquina MicroVAX 3100.

Con este proyecto se desea iniciar un proceso de estandarización de los Sistemas de Información de la Universidad de Juliaca a la nueva herramienta y además contar con una plataforma de Lenguaje de Cuarta Generación base que optimice las futuras mantenciones y también las consultas. Por ejemplo, en el sistema original para las consultas, es necesario realizar un programa que genere un listado con los registros consultados, el cual toma como mínimo un día. En cambio, con esta herramienta específicamente con los lenguajes interactivos de consultas (SQL y QUEL), esta tarea no demora más de 15 minutos.

La justificación de este proyecto de migración es que se

desean sofisticar los Sistemas de Información existentes en esta casa de estudios, con una herramienta que optimiza en gran medida el costo de tiempo de desarrollo de los sistemas, por consiguiente, la productividad en el desarrollo de software.

3.6.1.2 INVENTARIO DE COMPONENTES EXISTENTES

De acuerdo a lo recuperado del sistema existente (Sistema de Gestión Docente), se identificaron cinco clases de componentes existentes, las cuales son:

- Código fuente
- Archivos de documentación (manuales, etc)
- Archivos ejecutables y librerías de tiempo de ejecución
- Archivos de datos, pantallas e informes
- Componentes de hardware y software estándares.

En las tablas siguientes se muestran los componentes identificados en esta etapa con su ubicación física actual.

Ubicación física donde se encuentran los componentes inventariados (Lugar):

[1] Disco flexible

[2] Manual de sistema

[3] UANCV-S/34

[4] Manual de procedimientos

[5] Centro de cómputo

[6] sala de equipos

| CLASES | NOMBRE | DESCRIPCION DE COMPONENTES | LUGAR | ESTRAT |
|--------|--------|-----------------------------------|-------|--------|
| Código | ACTEST | Asignación de profesores | 3 | 1 |
| Fuente | ACTPRO | Actualiza profesores | 3 | 1 |
| | ACTCAR | Actualiza carreras | 3 | 1 |
| | ASISEL | Carga académica | 3 | 1 |
| | DECIDE | Resolver peticiones contrato | 3 | 1 |
| | DEPTOS | Mantener departamentos | 3 | 1 |
| | CONSSD | Solic.de servicios docentes dpto. | 3 | 1 |
| | CSSDES | Solic.de servicios docentes espec | 3 | 1 |
| | ACTFAC | Actualizar facultades | 3 | 1 |
| | PLAVIG | Mantiene curriculas vigentes | 3 | 1 |
| | CREGRU | Crear grupos | 3 | 1 |
| | CODORE | Consulta solicitud docentes dpto. | 3 | 1 |
| | CODOES | Consulta solicitud docentes espec | 3 | 1 |
| | COSODE | Consulta solicitud por dpto. | 3 | 1 |
| | COAGDE | Consulta secciones por dpto. | 3 | 1 |
| | COCDDE | Consulta carga docente por dpto. | 3 | 1 |
| | COCDFA | Consulta carga docente por facult | 3 | 1 |
| | INSORE | Listado solicitudes de dpto. | 3 | 1 |
| | INSOES | Listado solicitudes de especialid | 3 | 1 |
| | INPLVI | Listado de curriculas vigentes | 3 | 1 |
| | INPRDO | Listado provisión docente | 3 | 1 |
| ARCHI- | D.F.D. | Organigrama de procesos | 1,2 | 2 |
| VOS DE | D.D.P. | Organigramas descripción Procesos | 2 | 2 |
| DOCU- | D.D.A | Diagramas de descripción de Archi | 1,2 | 2 |
| MENTA- | A.M | Arbol de Menús | 2 | 2 |
| CION | F.P | Formularios de Pantallas | 2 | 2 |
| | F.I | Formularios de Informes | 2 | 2 |
| PROC. | M.P.A | Manual de procedimiento | 4 | 3 |
| ORGA. | | administrativo | | |
| Y ADMI | | | | |
| NISTRA | | | | |

ESTRATEGIAS:

- [1] El componente no es reutilizable
- [2] El componente tiene partes reusables
- [3] El componente satisface los objetivos de Ingeniería Reversa

| CLASES | NOMBRE | DESCRIPCION DE COMPONENTES | LUGAR | ESTRAT |
|---------------------------------|-------------------|--|---|--------|
| Componentes Software y Hardware | SDA | Generador de pantalla | 3 | 1 |
| | RLU | Generador de reportes | 3 | 1 |
| | SAM | Sistema administrador de menus | 3 | 1 |
| | SO | Sistema operativo OS/34 | 3 | 1 |
| | L.C | Lenguaje Cobol | 3 | 1 |
| | L.D.A | Libreria descripción de archivos | 3 | 1 |
| | | Equipo IBM SISTEMA/34 | 6 | 1 |
| | | Impresora | 5 | 1 |
| | | Terminales (6) | 5 | 1 |
| | Archivos de Datos | ASIGNA | Mantiene las asignaturas impartidas por los diferentes dptos. | 3 |
| DEPTOS | | Mantiene los departamentos existentes en la universidad | 3 | 2 |
| DOCENC | | Mantiene información sobre las asignaciones de profesores a los grupos | 3 | 2 |
| FACULT | | Mantiene las facultades existentes en la universidad | 3 | 2 |
| GRUPOS | | Mantiene los grupos formados en el semestre en curso de asignatur | 3 | 2 |
| PETICI | | Mantiene solicitudes de servicios docentes de departamento | 3 | 2 |
| PETESP | | Mantiene solicitudes de servicios docentes de especialidad | 3 | 2 |
| PROFES | | Mantiene los académicos tanto de planta, horas y contrata | 3 | 2 |
| | INSCRI | Mantiene información de alumnos inscritos en determinado curso | 3 | 2 |

| CLASES | NOMBRE | DESCRIPCION DE COMPONENTES | LUGAR | ESTRAT |
|-------------|-----------------------------------|-----------------------------------|-------|--------|
| Archivos | AC01 | Asignación de profesores | 3 | 1 |
| | AC02 | Actualiza profesores | 3 | 1 |
| Ejecutables | AC0301 | Actualiza carreras | 3 | 1 |
| | AC0302 | Carga académica | 3 | 1 |
| | AC0401 | Resolver peticiones para contrato | 3 | 1 |
| | AC0402 | Mantener departamentos | 3 | 1 |
| | | Módulo de servicios | | |
| | SE0101 | Solic.de servicios docentes espec | 3 | 1 |
| | SE0102 | Actualizar facultades | 3 | 1 |
| | SE03 | Mantiene curriculas vigentes | 3 | 1 |
| | | Módulo de consultas | | |
| | CO0101 | Consulta solicitud docentes espec | 3 | 1 |
| CO0102 | Consulta solicitud por depto. | 3 | 1 | |
| CO0103 | Consulta grupos por dpto. | 3 | 1 | |
| CO0104 | Consulta carga docente por dpto. | 3 | 1 | |
| CO0201 | Consulta carga docente por facult | 3 | 1 | |
| CO0202 | Listado solicitudes de dpto. | 3 | 1 | |
| | Módulo de informes | | | |
| IN01 | Listado solicitudes de dpto. | 3 | 1 | |
| IN02 | Listado solicitudes de especialid | 3 | 1 | |
| IN03 | Listado curriculas vigentes | 3 | 1 | |
| IN04 | Provisión docente | 3 | 1 | |
| | Librerías software de apoyo | | | |
| SDA | Generador de pantallas | 3 | 1 | |
| RLU | Generador de reportes | 3 | 1 | |
| SAM | Sistema administrador de menues | 3 | 1 | |
| SEU | Editor de programas | 3 | 1 | |
| CL | Lenguaje de control | 3 | 1 | |

3.6.1.3 DEFINIR LA ESTRATEGIA DE INGENIERIA REVERSA

En esta etapa se desarrolla un análisis global de calidad y reusabilidad de los componentes identificados.

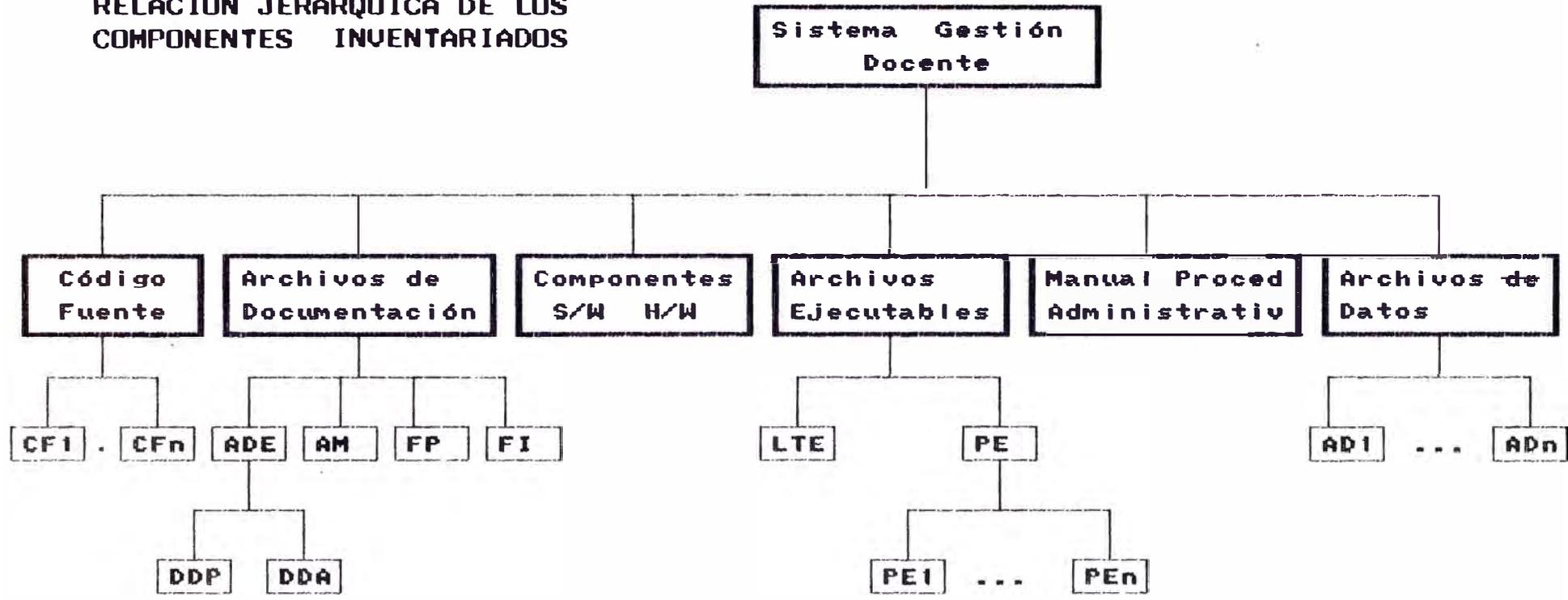
A continuación, se muestran en la figura 3.2 la relación jerárquica existente entre los componentes del sistema de Gestión Docente identificados.

Esta estructura jerárquica permite mostrar la relación "es-parte-de" entre los componentes identificados y sus respectivas clases. Además junto con el análisis de calidad y reusabilidad, mostrar los niveles de aceptación dentro de cada clase para las futuras decisiones que se tomen con respecto al reuso de los componentes.

Los atributos que permiten tomar una decisión con respecto a la estrategia justificada para cada componente inventariado, son los siguientes:

- **El atributo de apoyo:** Este atributo mide el grado de disponibilidad del componente, es decir, si se encuentra disponible en que forma, ya sea electrónicamente o manualmente.
- **El atributo de estructura:** Este atributo mide el grado de estructura del componente, es decir, si existe un bajo grado de estructura es necesario leer completamente el documento. En caso contrario bastará una visión menos detallada y si es posible con herramientas automáticas ad-hoc, lo que involucra un menor costo de tiempo.
- **El atributo de relevancia:** Este atributo mide el grado de contribución del componente con los objetivos fijados para la aplicación, es decir, si un objeto no es relevante a los objetivos del proyecto se descartará sin afectar la funcionalidad del sistema. En caso contrario, si contribuye a

RELACION JERARQUICA DE LOS COMPONENTES INVENTARIADOS



CF1-n : Código Fuente 1 - n

ADE : Archivo de Documentación Estructurado

DDP : Diagrama de Descripción de Procesos

DDA : Diagrama de Descripción de Archivos

AM : Arbol de Menues

FP : Formularios de Pantallas

FI : Formularios de Informes

LTE : Librería de Apoyo

PE1-n : Programas Ejecutables 1 - n

AD1-n : Archivos de Datos 1 - n

FIGURA 3.2

los objetivos planteados, entonces este componente es esencial en el correcto funcionamiento u operación de sistema, por consiguiente, para su calidad.

De acuerdo a estos tres atributos de calidad, se definirán tres estrategias básicas, las cuales serán aplicadas sobre cada uno de los componentes inventariados. A partir de estas estrategias se elabora un plan de actividades que en la etapa siguiente de este proceso se desarrollará.

Las tres estrategias básicas que serán utilizadas en este análisis son las siguientes:

[1] El componente no es utilizable, y será descartado; la estrategia de Ingeniería Reversa es reconstruir este componente directamente en el depósito, excepto para el código fuente, el cual se recodificará en el nuevo ambiente de desarrollo y en la fase de Reimplementación.

[2] El componente contiene partes reusables, las cuales pueden ser analizadas y extraídas; la estrategia especificará la técnica que será utilizada para extraer las partes del componente.

[3] El componente satisface los objetivos de Ingeniería Reversa, y puede ser tomado sin modificaciones. La estrategia es transferir este componente al depósito.

En la tabla anterior, específicamente en la última columna (ESTRATEGIA) se detallan las estrategias que se aplicarán a los diferentes componentes inventariados.

Para la clase de componentes código fuente no cumple el objetivo principal de estandarización de los sistemas de la Universidad de Juliaca y más específicamente el objetivo de migración de este sistema a la herramienta de desarrollo de sistemas de información Ingres. Por esta razón se decidió por

la estrategia número 1. Además, la máquina Micro Vax 3100 no incluye compilador Cobol, por lo tanto, sería imposible considerar la posibilidad de reusar los códigos fuentes del sistema existente.

Más del 90% de la clase de archivos de documentación tiene un nivel pobre de calidad, por ende, la estrategia utilizada es la número 2, que extrae las partes reusables del componente para el nuevo sistema de Gestión Docente a reimplementar. Al porcentaje menor se le aplicará la estrategia número 3, es el caso del archivo de documentación Arbol de Menús, porque existían herramientas como el SAM que permitían tener un buen nivel de documentación de éstos, sólo se incluyeron algunos nuevos procesos que son necesarios para la implementación.

Para la clase de archivos ejecutables se aplica la estrategia número 1, igual a la aplicada al código fuente.

Para la clase de componentes de software y hardware, la cual depende directamente sobre los recursos proporcionados por la máquina a la cual se desea migrar y también sobre el ambiente de desarrollo del sistema. Para esta aplicación los componentes software no satisfacen los objetivos perseguidos, por consiguiente, la estrategia fijada es la número 1, es decir, descartar los recursos anteriores y utilizar los recursos proporcionados por la nueva máquina y el ambiente de desarrollo. Para el caso de componentes hardware se fijaron las estrategias 1 y 2 independientemente, es decir, la máquina sobre la cual se explota el sistema existente (S/34 IBM) se descarta y se utiliza la máquina en la cual se desea migrar el sistema (MicroVax 3100) para el caso de los otros componentes hardware, la estrategia fijada fue la de reutilizar éstas en lo posible.

Para la clase de archivo de datos se fijó la estrategia número 2, porque se reutilizará la estructura y los datos de

estos archivos. Los datos se reutilizarán para poblar la Base de Datos que se generará como resultado de la fase de Reimplementación.

3.6.2 ETAPA DE EJECUCION DE LAS ACTIVIDADES DE INGENIERIA REVERSA

De acuerdo al plan elaborado anteriormente, en esta etapa corresponde realizar todas las acciones planificadas, utilizando las actividades detalladas en el capítulo I, o combinaciones de éstas, todo esto con el objeto de cargar el depósito de información, el cual al finalizar esta etapa será una fiel representación del sistema de Gestión Docente que se desea reimplementar. A partir de este depósito de información se captura las especificaciones y el diseño para realizar la fase de Reimplementación del sistema en cuestión.

3.6.2.1 DIAGRAMA DE ENTIDAD RELACION

Dentro de la estrategia de Ingeniería Reversa en el atributo de apoyo se encontró que no existía en forma manual ni electrónica un documento muy importante que permite entender las relaciones que existen entre las estructuras de datos del sistema más conocido como diagrama de Entidad/Relación.

Uno de los enfoques más usados para resolver el problema de diseñar Base de Datos es el Diagrama de Entidad/Relación, que es una técnica para representar gráficamente la estructura lógica de una Base de Datos, con lo cual se ofrece una forma sencilla y comprensible de comunicar los rasgos prominentes del diseño de cualquier Base de Datos.

En los atributos de estructura y relevancia solo se puede mencionar que el Diagrama de Entidad/Relación tiene un alto

grado de contribución para aportar primeramente con una documentación que nos permita lograr el objetivo general, la cual es, la estandarización de los Sistemas de Información de la U.A.N.C.V.

Para lograr el Diagrama de Entidad/Relación primeramente identificamos todos los archivos con los cuales trabajaba el sistema: Asignatura, Facultad, Parámetros, Inscripciones, Profesores, Carreras, Currícula, Departamento, Grupos, Petición Departamento, Docencia, Petición Especialidad, Secciones, Petición Contratados; con los cuales se lograban los diferentes listados Después de una serie de procesos automáticos y manuales.

En el Diagrama de Entidad/Relación, se muestra la relación de pertenencia que existe entre la entidad Facultad y el Departamento, una Facultad puede tener un departamento o no, ya que algunas Facultades no tienen un departamento por ejemplo: las Facultades de Derecho, Medicina Veterinaria, etc, y un departamento pertenece a una Facultad, ya que esta no puede existir independiente de una Facultad -El Departamento de Ciencias pertenece a la Facultad de Ingenierías y el Departamento de letras a la Facultad de Educación- según lo estipulado en el estatuto de la universidad; además la Facultad puede tener una o varias carreras, se utiliza el término "puede" porque hay facultades que solamente tienen una sola carrera, ejemplo la Facultad de Derecho y otras pueden tener mas de una carrera académico profesional, ejemplo la Facultad de Ingenierías tiene cuatro carreras académico profesionales, y una carrera pertenecer a una Facultad.

En el párrafo anterior se describe el tipo de relación que existe entre las entidades FACULTAD, DEPARTAMENTO Y CARRERAS, y de igual manera se deberá leer el Diagrama de Entidad/Relación, mostrado en la documentación del presente capítulo.

3.6.2.2 DIAGRAMA DE FLUJO DE DATOS

Se hizo un análisis cuantitativo y cualitativo de los organigramas de descripción de procesos para poder conocer la calidad de estos elementos, además analizamos el rol de estos componentes en el sistema, para ello se contrastaron contra los programas fuente existentes detectándose que no todas las correcciones hechas a los programas se reflejaban en dichos organigramas.

Para ilustrar mejor lo dicho anteriormente, en la figura 3.3, mostramos el organigrama del proceso provisión docente.

Debido a que el Organigrama original carecía de un pseudocódigo que nos refleje la lógica de descripción del proceso y la lógica de programación y manteniendo el objetivo final para este organigrama se hizo un análisis de Caja Negra al proceso de Provisión docente, que consistió en hacer una serie de pruebas suministrándole diferentes grupos de datos de entrada que el programa requería, obtener y estudiar los resultados para cada grupo de datos; así se pudo determinar de que algunos datos de entrada que exigía el programa no eran utilizados para generar la información ni como salidas, por lo cual se optó por no considerarlos y el sistema siguió arrojando los mismos resultados; además se encontró que muchos datos que eran suministrados manualmente desde el teclado eran parte del proceso, y estos mismos datos se ingresaban en otros procesos por ello creamos archivos con estos datos. Con todo esto se pudo elaborar el pseudocódigo de este proceso.

Para el atributo de relevancia la contribución del Proceso de Provisión Docente a los objetivos fijados era bueno pero de todas maneras demandaba que se automatizaran partes que se llevaban en forma manual.

Según las tres estrategias de la Ingeniería Reversa a aplicar a cada uno de los elementos, luego de estudiar sus

DIAGRAMA DEL PROCESO

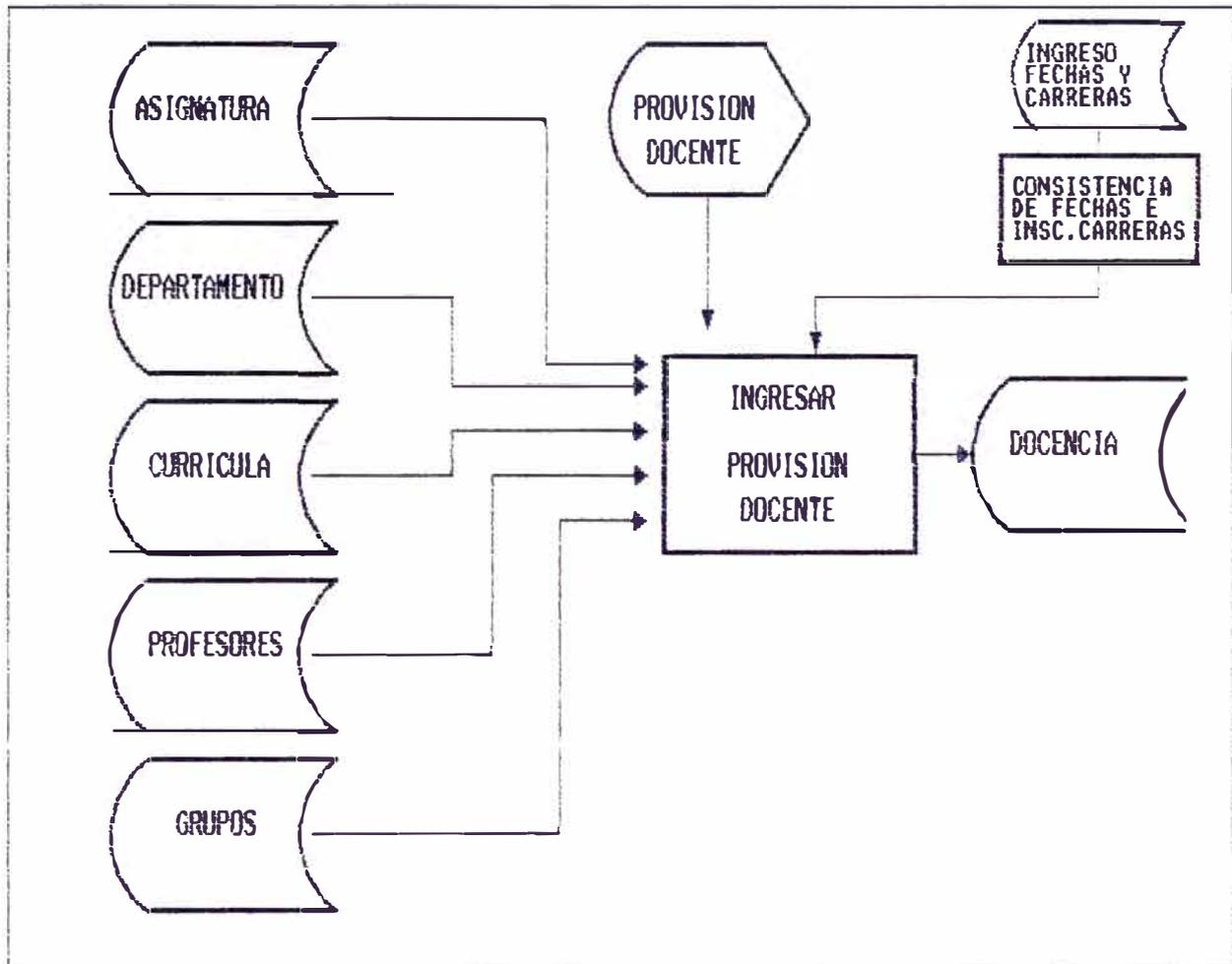


FIGURA No 3.3

diferentes atributos, este cae en la estrategia número 2, porque contiene partes reusables las cuales fueron analizadas y extraídas. Como resultado se obtuvo el Diagrama de Proceso Provisión Docente mostrado en el Anexo 3 con su respectivo seudocódigo.

De igual manera se logró revisar los demás Organigramas de Proceso y para cada uno se tuvo que desarrollar los mismos pasos que para el proceso de provisión docente, y así obtener su seudocódigo.

3.6.2.3 DIAGRAMAS DE DESCRIPCION DE ARCHIVOS

En la mayoría de los diagramas de descripción de archivos se tenía una relación de campos pero no se tenía su clave de acceso, además algunos campos habían sido eliminados y otros añadidos los cuales no se reflejaban en la documentación de diseño.

Fue arduo recopilar las verdaderas descripciones de archivos ya que en los programas solo se mencionaban los campos que este utilizaba, entonces se tuvo que revisar todos los programas y contrastar con los archivos físicos y formar de esta manera la estructura de los archivos.

Debido a que no existía una estandarización en el ingreso de algunos datos, como las fechas, detectamos que en varios archivos se encontraron grupos de registros con el formato Día Mes Año, debiendo ser Año Mes Día, lo cual ocasionaba que se obtuvieran listados con cierto grado de inconsistencia cuando se hacían procesos en función a la fecha.

Otro de los problemas detectados fue que algunos campos por la naturaleza de su estructura, ejemplo la fecha, en algunos programas se definían como numéricos y en otros como alfanumérico de tal manera cuando un proceso coincidía en

utilizar estos campos definidos en diferentes archivos se cancelaba el programa, por hacer una comparación lógica o matemática entre campos de diferente tipo.

Para una mejor ilustración de lo antes mencionado tomemos como ejemplo el archivo Petición Departamento, del cual se encontró en su documentación lo siguiente:

ARCHIVO PETICION-DPTO

| Campo | Tipo | Longitud |
|--------------|------|----------|
| Cod-Pet-Dpto | X | 11 |
| Cod-Fac | 9 | 1 |
| Correl-Car | 9 | 2 |
| Anno | 9 | 2 |
| Cod-Dpto | X | 3 |
| Correl-Asig | 9 | 3 |
| Num-Alum | 9 | 3 |
| Catedra | 9 | 2 |
| Practica | 9 | 2 |
| Laborat | 9 | 2 |
| F-Ing-Pet | 9 | 6 |
| Seccion | X | 1 |
| Tipo-Asig | X | 2 |

Este archivo es encontrado en forma manual. Como se puede observar los nombres de los campos no tienen una descripción que permita identificar su función, como por ejemplo el campo ANNO no podemos saber a que año se refiere, año de la curricula, año de ingreso, etc. Luego de revisar los programas y hacer algunas pruebas determinamos que en la documentación faltaban los campos :

| CAMPO | TIPO | LONGITUD |
|---------------|------|----------|
| Num-alu-Real | 9 | 3 |
| Catedra-Real | 9 | 2 |
| Practica-Real | 9 | 2 |
| Laborat-Real | 9 | 2 |

La gran mayoría de campos tenían la misma descripción (nombre, longitud y tipo) de la documentación trabajando en el sistema.

Este archivo tenía entre otros el campo TIPO-ASIG que carecía de significado lo cual nos llevó a revisar el contenido de este con mayor detenimiento, descubriéndose que sólo en los primeros registros tenían datos que nunca se utilizaron e investigando al respecto concluimos que tampoco se utilizarían en el futuro; por lo cual se procedió a calificar a este campo con la estrategia número 3.

El atributo de estructura del archivo era bajo por lo cual se tuvo que revisar los programas para obtener la estructura completa del archivo. El grado de contribución de este componente fue relevante para el logro de los objetivos de sistema de Gestión Docente ya que este era esencial para el correcto funcionamiento del Sistema.

Utilizando las estrategias de Ingeniería Reversa calificamos a este elemento como 2, ya que se tuvo que analizar y extraer de los programas la estructura de datos, de igual manera se procedió con los demás archivos del sistema.

3.6.2.4 ARBOL DE MENÚES

El árbol de menús en contraste con la demás documentación reflejaba las correcciones y ampliaciones hechas al Sistema de Gestión Docente.

El grado de relevancia del árbol de menús es necesario para el mejor entendimiento del sistema por lo cual se contaba con este elemento ajustado al sistema que estaba trabajando, ya que se utilizaba un sistema administrador de menús que le permitía reflejar en los menús toda adición o eliminación de opciones.

Esta documentación se encontraba en forma electrónica la cual sirvió para ajustar pequeños cambios y obtener como resultado el documento que se muestra en el anexo 3.

3.6.3 ETAPA DE CONSOLIDACION Y VERIFICACION DE LOS RESULTADOS

En esta etapa se toma el depósito de información obtenido a través de las consecutivas etapas de esta metodología y se verifica su consistencia y totalidad. Esto se realiza revisando que cada componente inventariado tenga su representación en este depósito y así chequear la totalidad de esta información. Además se verifica la consistencia tomando todos los objetos participantes en la documentación, revisando que cada uno de ellos representen fielmente el objeto (estructuras de datos, archivos de datos, procesos, jerarquía de menús, entidades participantes, relaciones, interacciones) dentro del sistema.

3.7 DOCUMENTACION DEL SISTEMA DE INFORMACION: GESTION DOCENTE

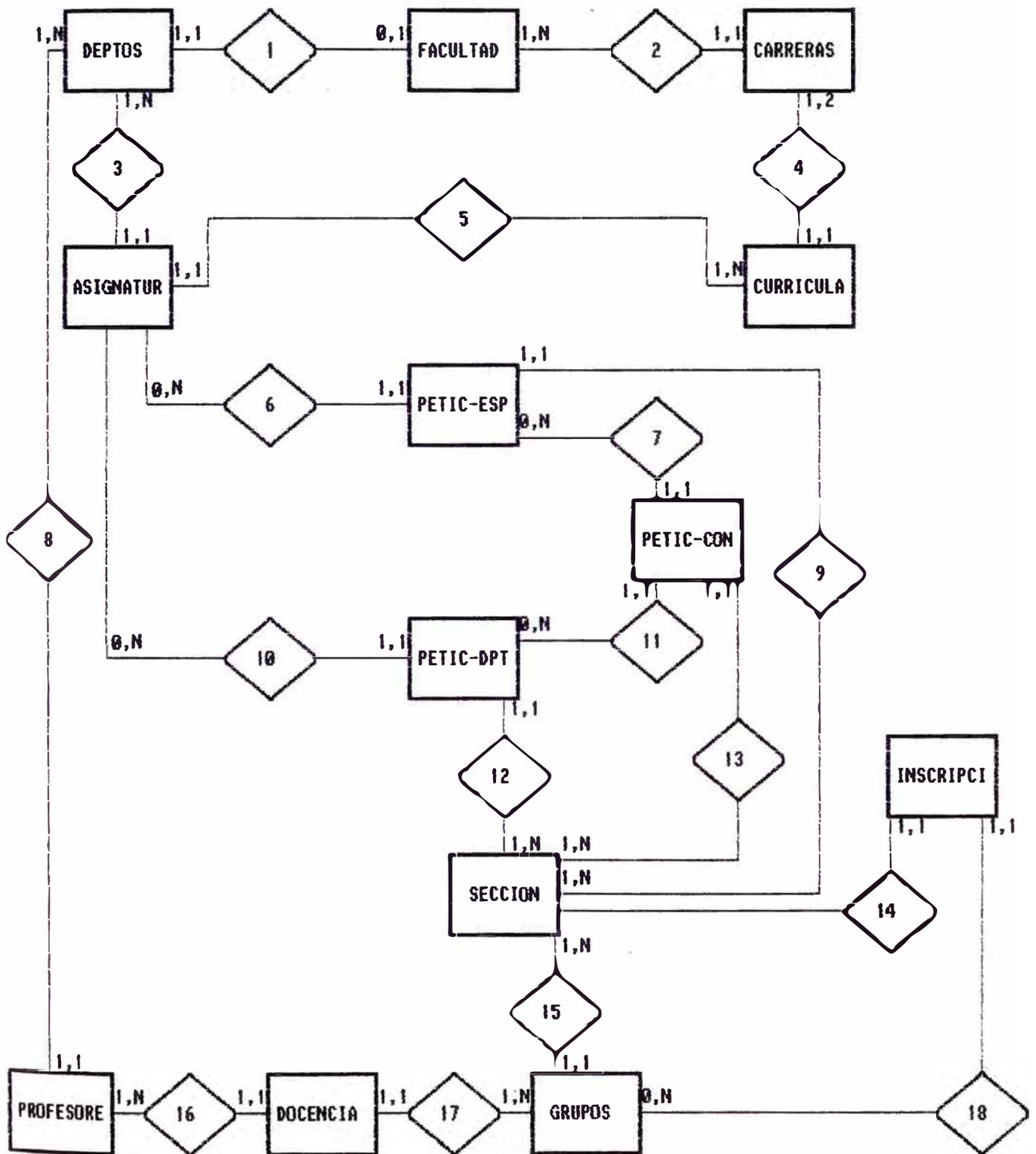
A continuación, se incluye la documentación resultante del Sistema de Información Gestión Docente, la cual se obtuvo a través de las sucesivas etapas de la metodología de Ingeniería Reversa. La documentación, se obtuvo en base a la escasa documentación existente de el sistema anterior.

Esta documentación es la representación confiable del sistema, por lo tanto, es una buena base para cualquier mantención futura sobre este sistema.

La documentación detallada en este anexo está estructurado en base a la metodología de Análisis y Diseño Estructurado.

También se incluye el Modelo de Entidad-Relación, una tabla de descripción de las relaciones entre los procesos versus los archivos y el árbol de menues del sistema.

DIAGRAMA DE ENTIDAD-RELACION



DESCRIPCION RELACIONES

IDENTIFICACION

SIGNIFICADO



PERTENECEN



TIENE



DICTAN



POSEE



SON-DE



GENERA



GENERA



SUPLE



FORMAN



SON-DE

DESCRIPCION RELACIONES

IDENTIFICACION

SIGNIFICADO



GENERA



FORMAN



FORMAN



AFECTAN



DAN-ORIGEN



TIENE



CORRESPONDE



AFECTAN

DIAGRAMA DE CONTEXTO

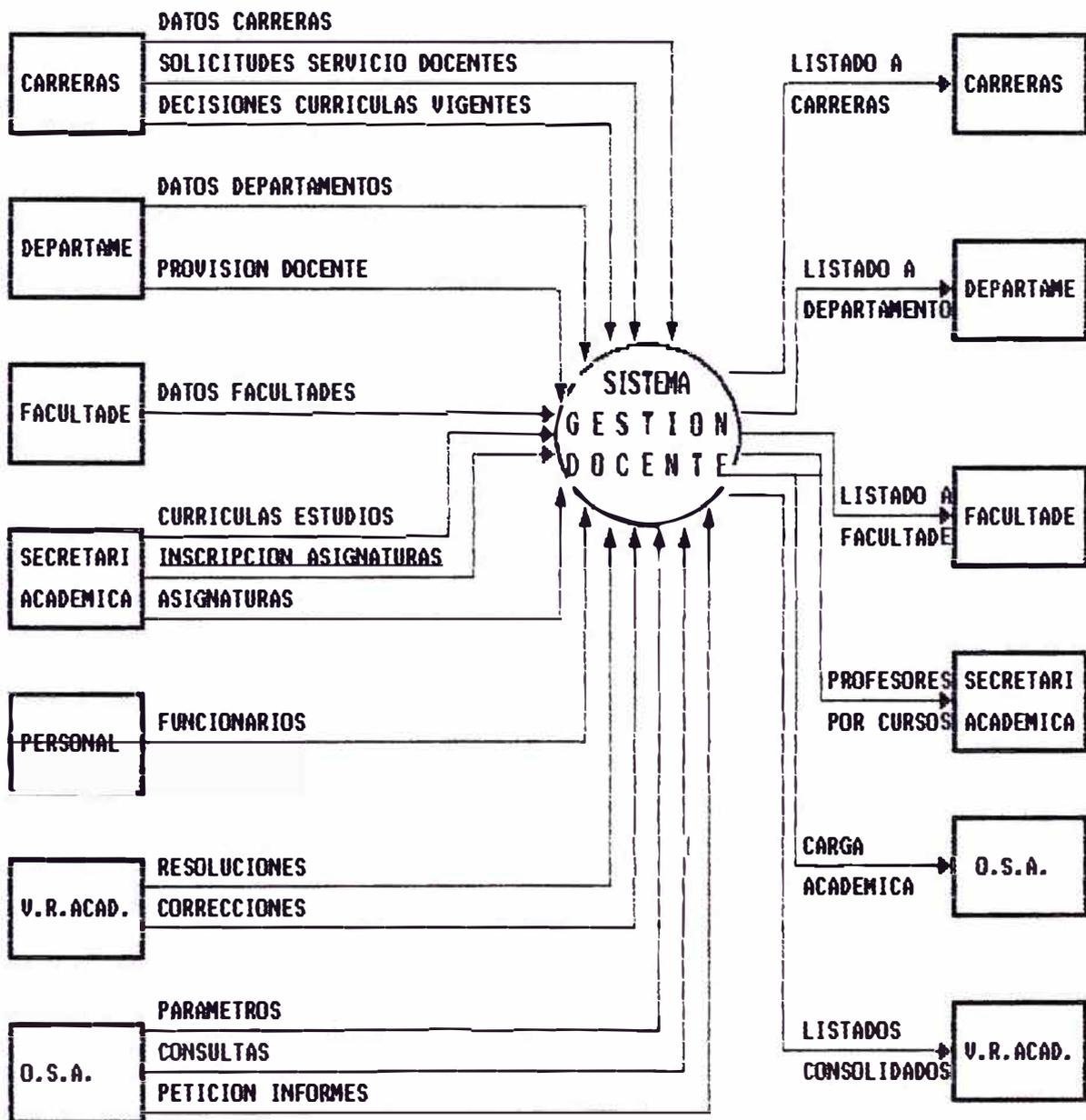
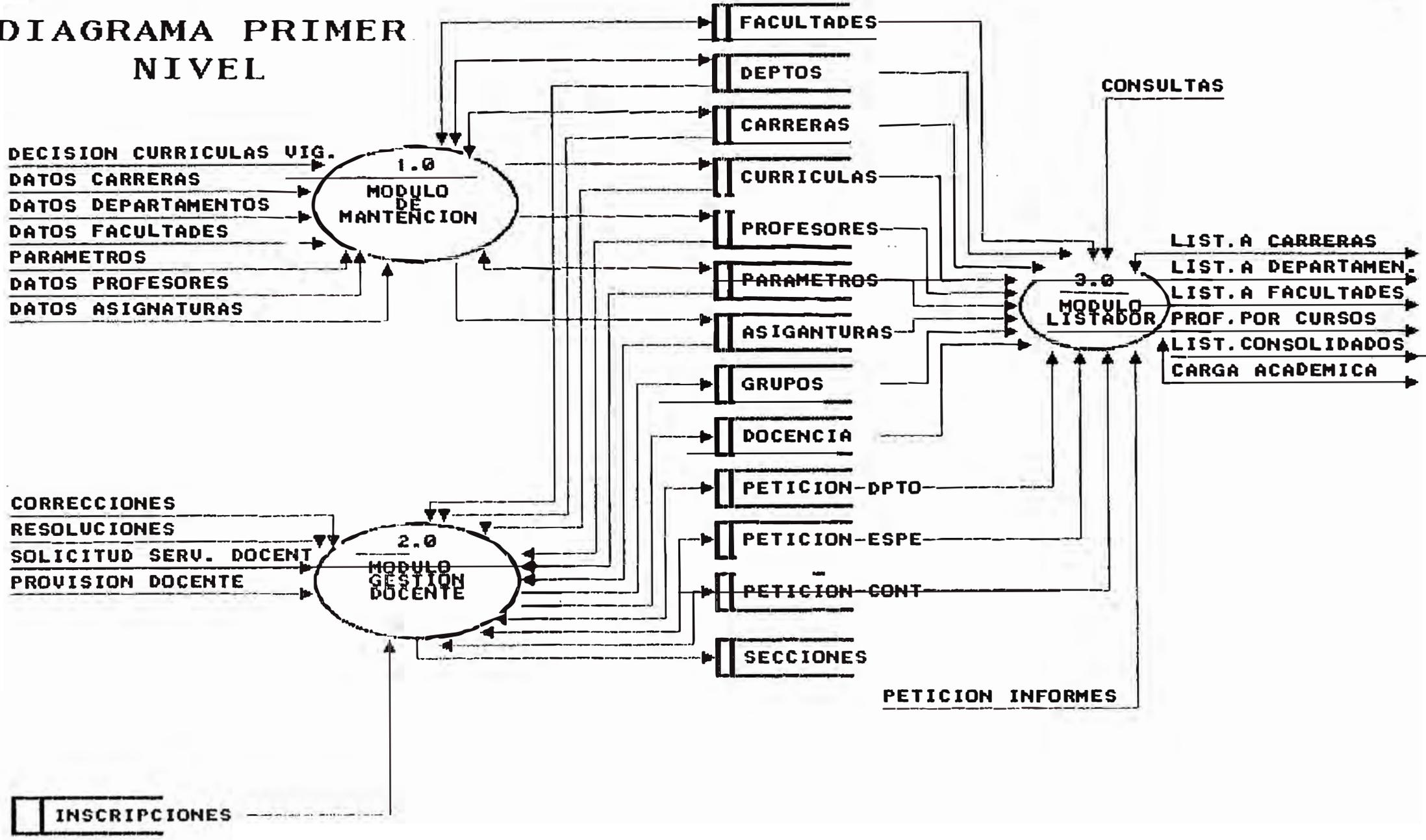
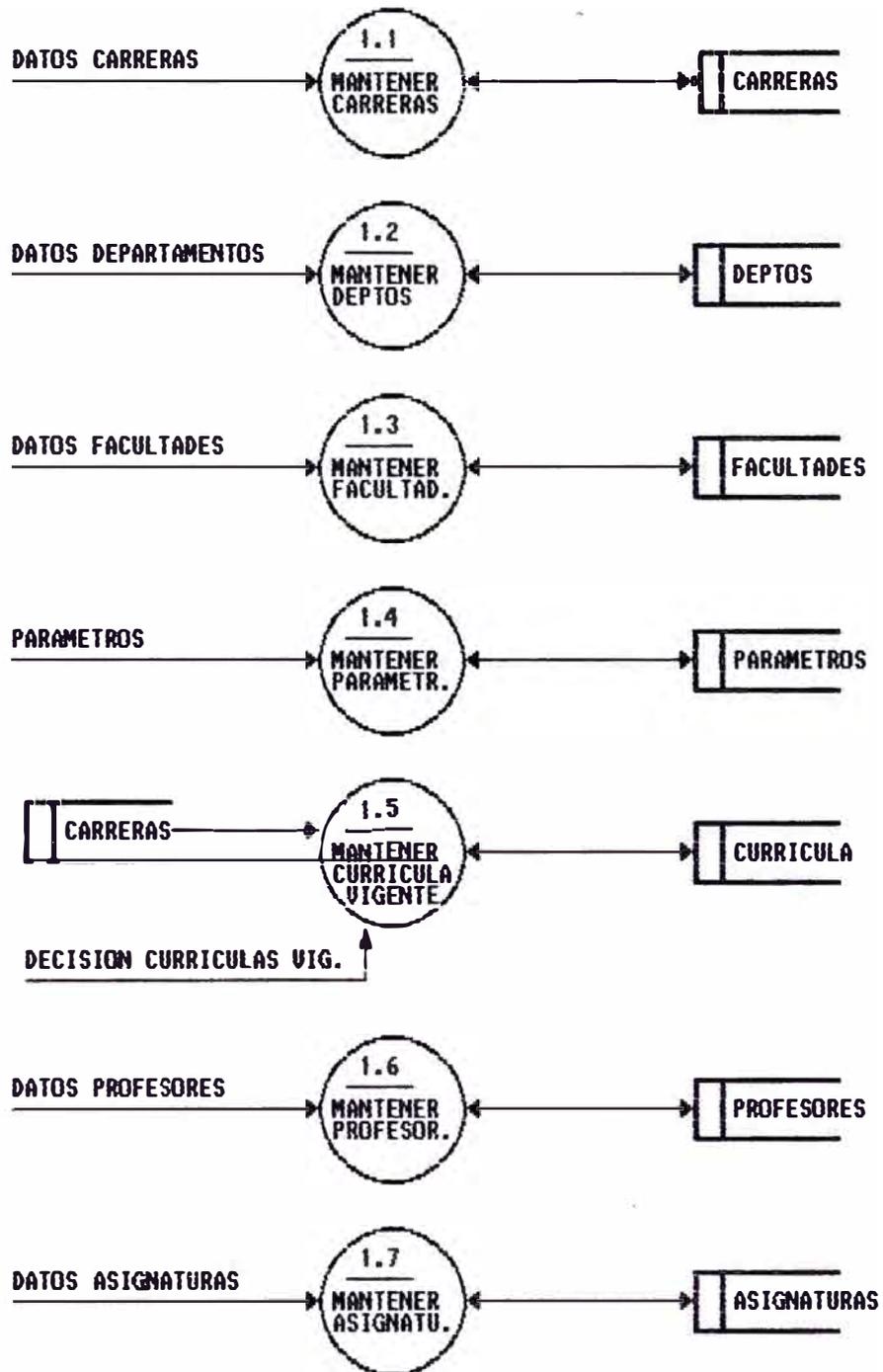


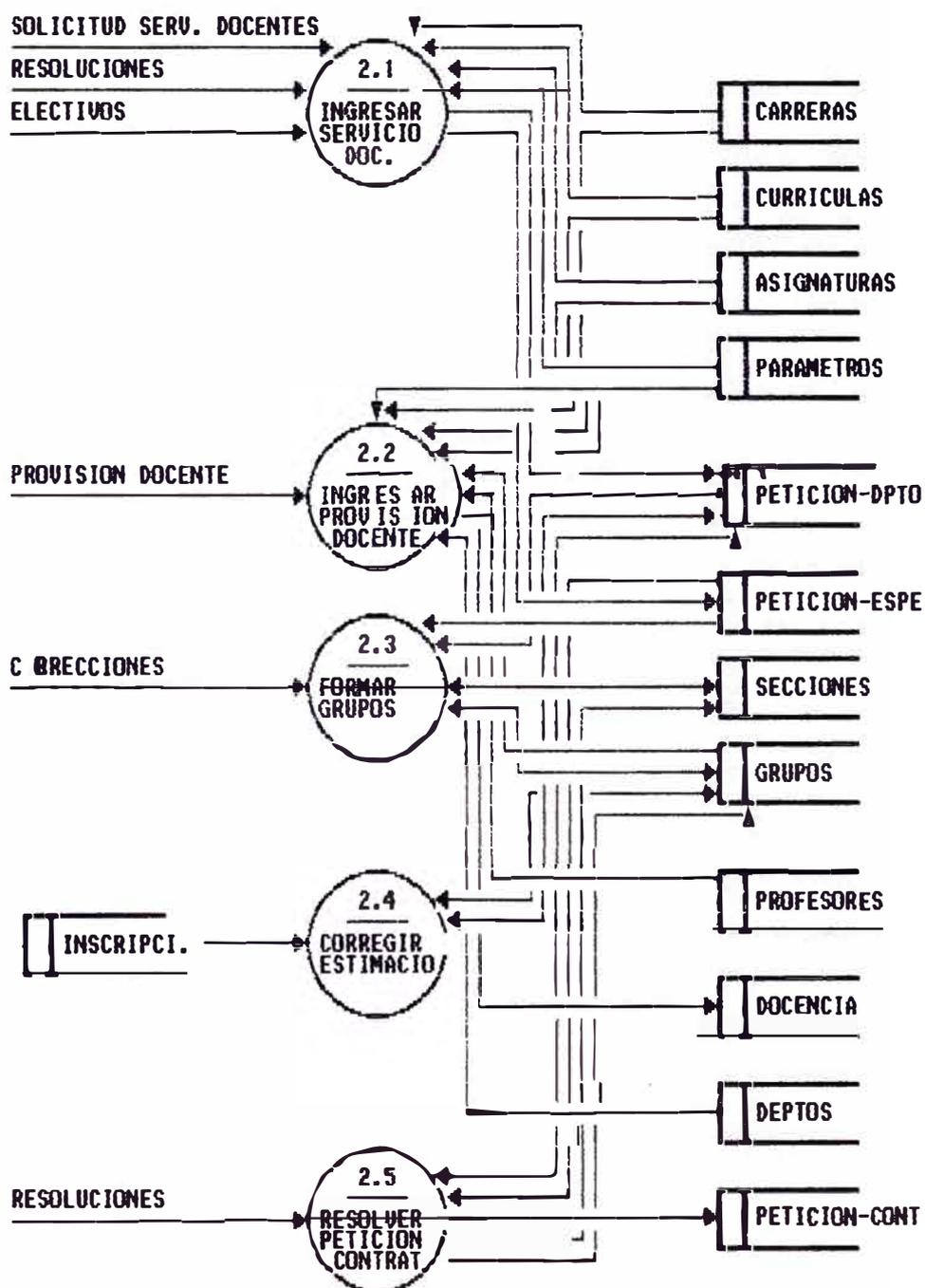
DIAGRAMA PRIMER NIVEL



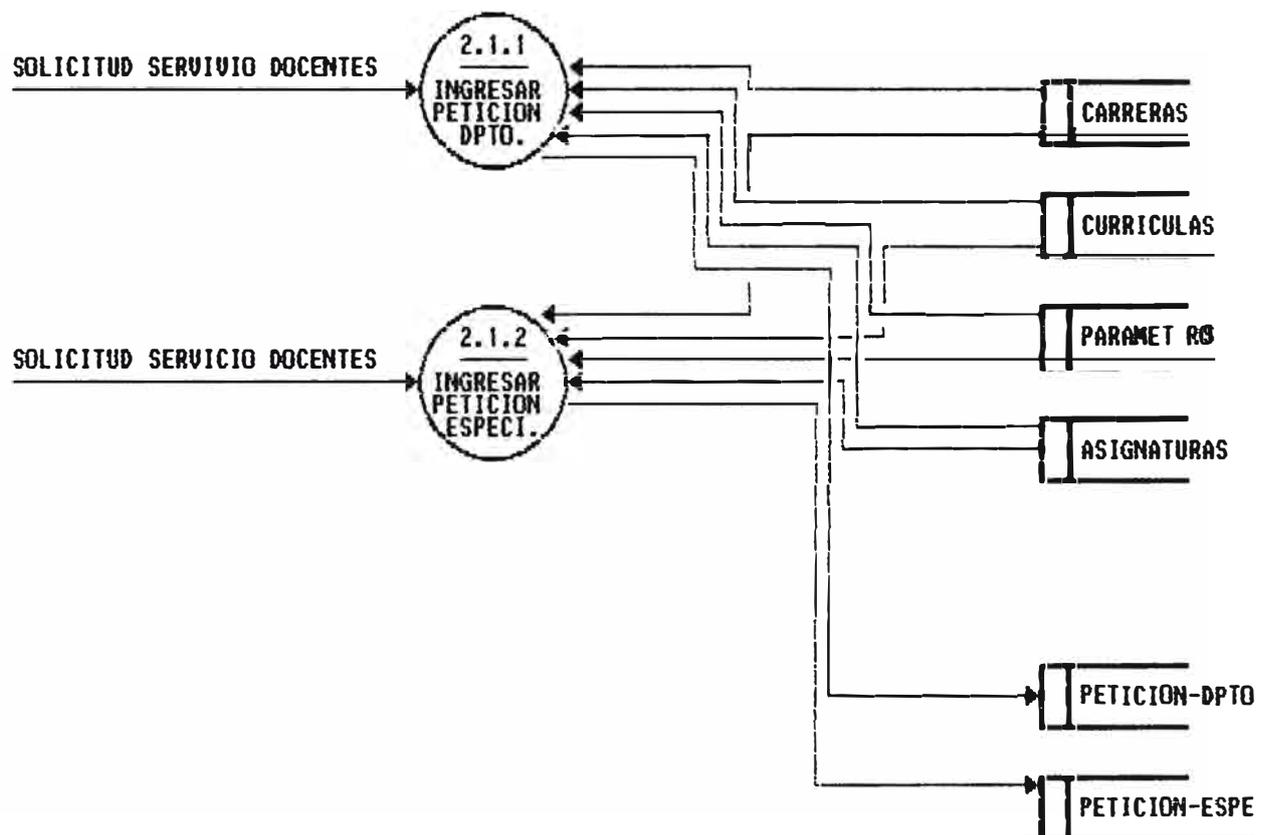
EXPLOSION DE BURBUJA 1.0



EXPLOSION BURBUJA 2.0



EXPLOSION BURBUJA 2.1



EXPLOSION BURBUJA 2.3

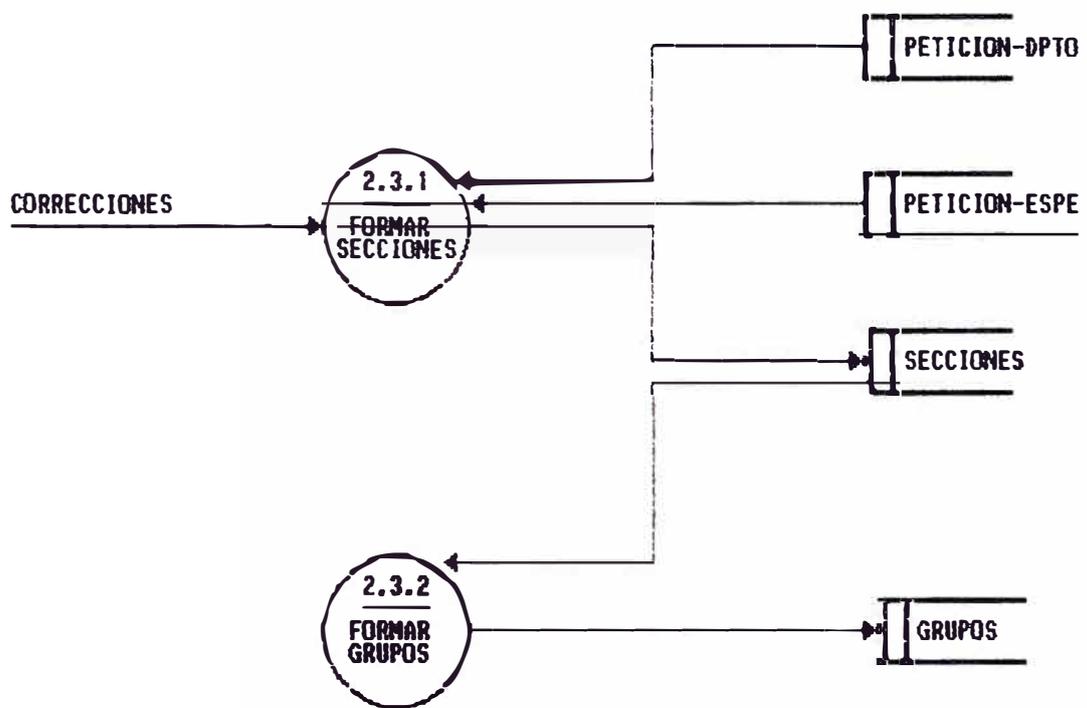


DIAGRAMA DEL PROCESO



REGLAS DEL PROCESO

```
INICIO
  ABRIR ARCHIVO FACULTADES
  INGRESAR OPCION
  MIENTRAS OPCION <> FIN HACER
    SI OPCION = INGRESAR
      INGRESAR FACULTADES
    SI OPCION = ELIMINAR
      ELIMINAR FACULTADES
    SI OPCION = MODIFICAR
      MODIFICAR FACULTADES
    EN CASO CONTRARIO
      ERROR
  INGRESAR OPCION
FIN-BUCLE
CERRAR ARCHIVO FACULTADES
FIN
```

DIAGRAMA DEL PROCESO



REGLAS DEL PROCESO

```
INICIO
  ABRIR ARCHIVO DEPARTAMENTOS
  INGRESAR OPCION
  MIENTRAS OPCION (<) FIN HACER
    SI OPCION = INGRESAR
      INGRESAR DEPARTAMENTOS
    SI OPCION = ELIMINAR
      ELIMINAR DEPARTAMENTOS
    SI OPCION = MODIFICAR
      MODIFICAR DEPARTAMENTOS
  EN CASO CONTRARIO
    ERROR
  INGRESAR OPCION
FIN-BUCLE
CERRAR ARCHIVO DEPARTAMENTOS
FIN
```

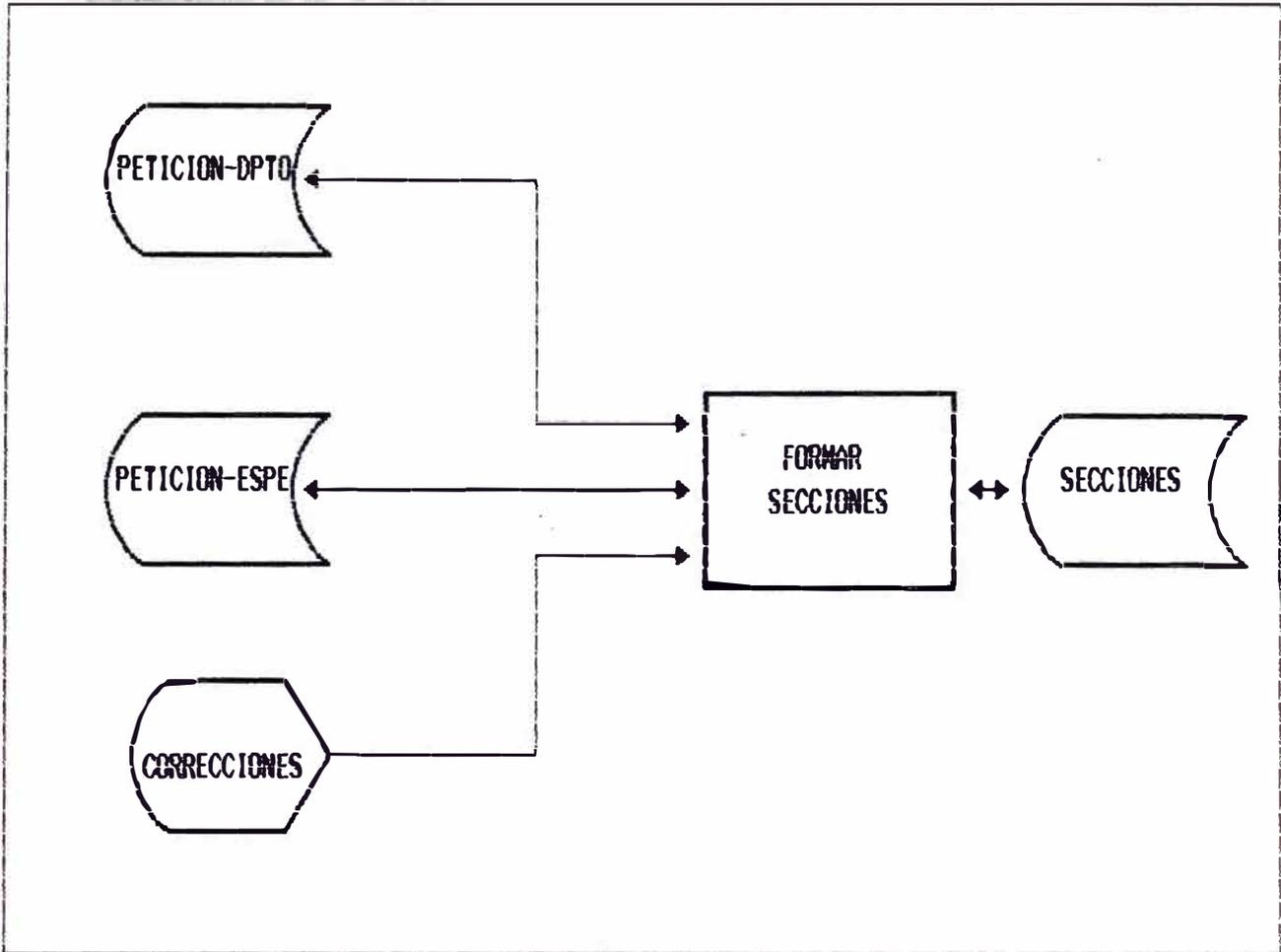
DIAGRAMA DEL PROCESO



REGLAS DEL PROCESO

```
INICIO
  ABRIR ARCHIVO CARRERAS
  INGRESAR OPCION
  MIENTRAS OPCION <> FIN HACER
    SI OPCION = INGRESAR
      INGRESAR CARRERAS
    SI OPCION = ELIMINAR
      ELIMINAR CARRERAS
    SI OPCION = MODIFICAR
      MODIFICAR CARRERAS
  EN CASO CONTRARIO
    ERROR
  INGRESAR OPCION
FIN-BUCLE
CERRAR ARCHIVO CARRERAS
FIN
```

DIAGRAMA DEL PROCESO



REGLAS DE PROCESO:

INICIO

ABRIR ARCHIVOS PETICION-DPTO, PETICION-ESPE, SECCIONES

LEER REGISTRO PETICION-DPTO, PETICION-ESPE

MIENTRAS NO EOF PETICION-DPTO, PETICION-ESPE HACER

SI DESEA CORREGIR

INGRESAR CORRECCIONES

FORMAR SECCIONES

CARGAR ARCHIVO SECCIONES

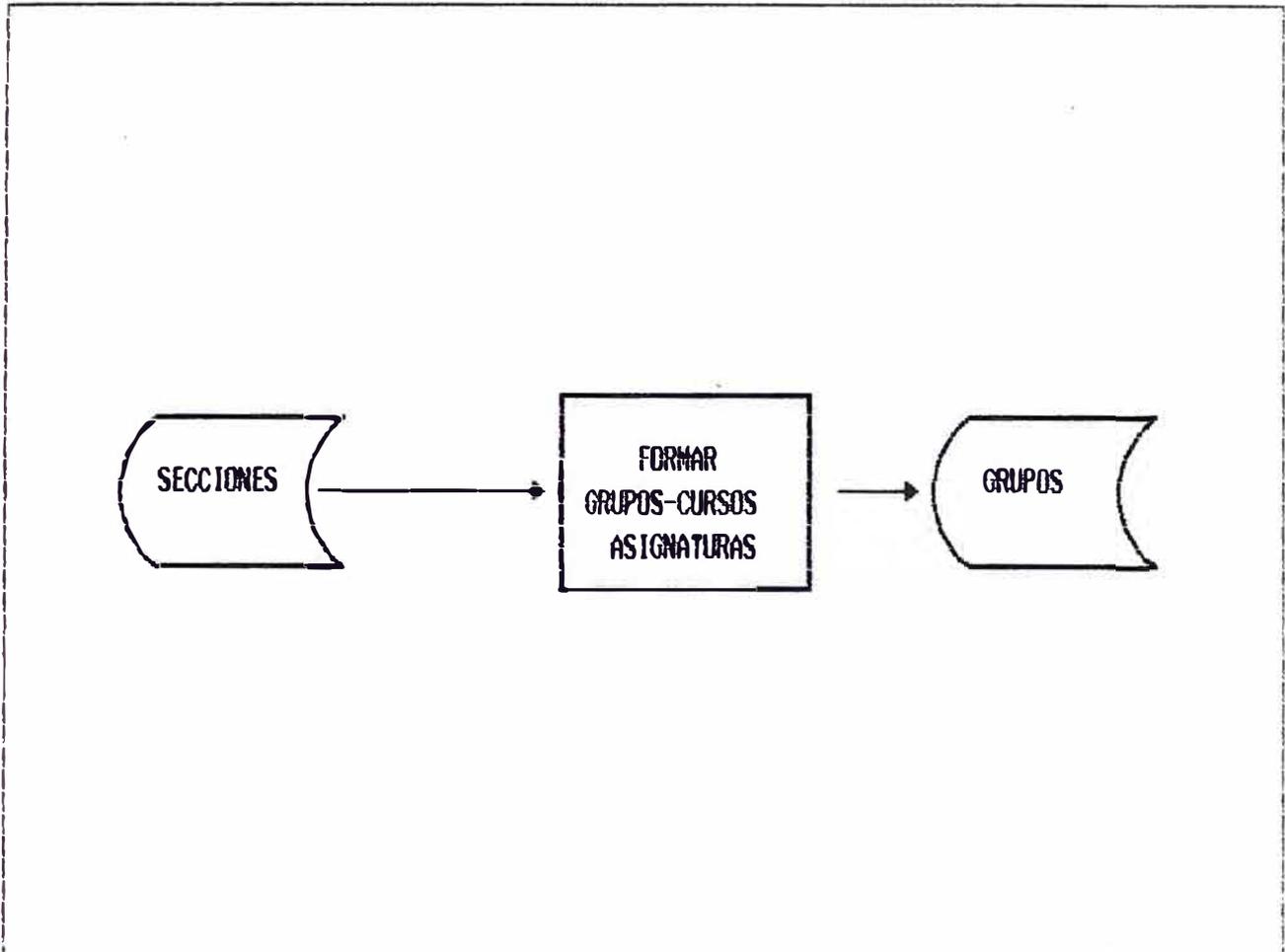
LEER REGISTRO PETICIONES-DPTO, PETICION-ESPE

FIN-BUCLE

CERRAR ARCHIVOS

FIN

DIAGRAMA DEL PROCESO



REGLAS DE PROCESO:

INICIO

ABRIR ARCHIVOS SECCIONES, GRUPOS

LEER REGISTRO SECCIONES

MIENTRAS NO EOF SECCIONES HACER

DESCOMPONER LA SECCION EN GRUPOS (CATEDRA, PRACTICA, LABORATORIO)

CARGAR ARCHIVO GRUPOS CON LAS DIVISIONES REALIZADAS

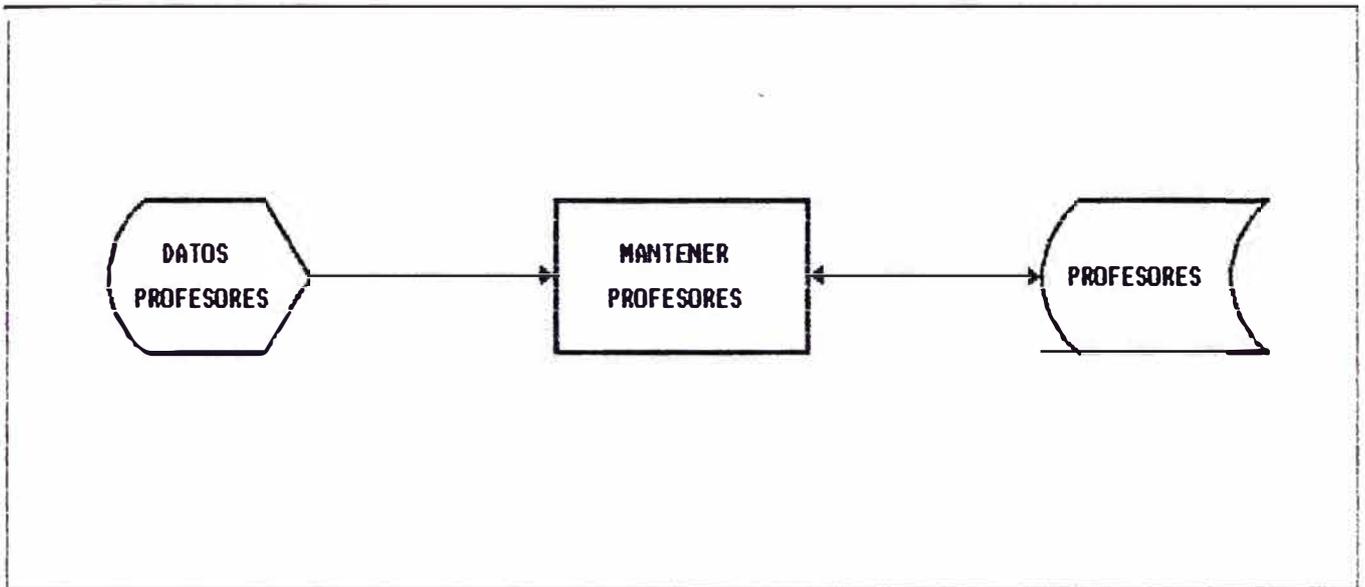
LEER REGISTRO SECCIONES

FIN-BUCLE

CERRAR ARCHIVOS

FIN

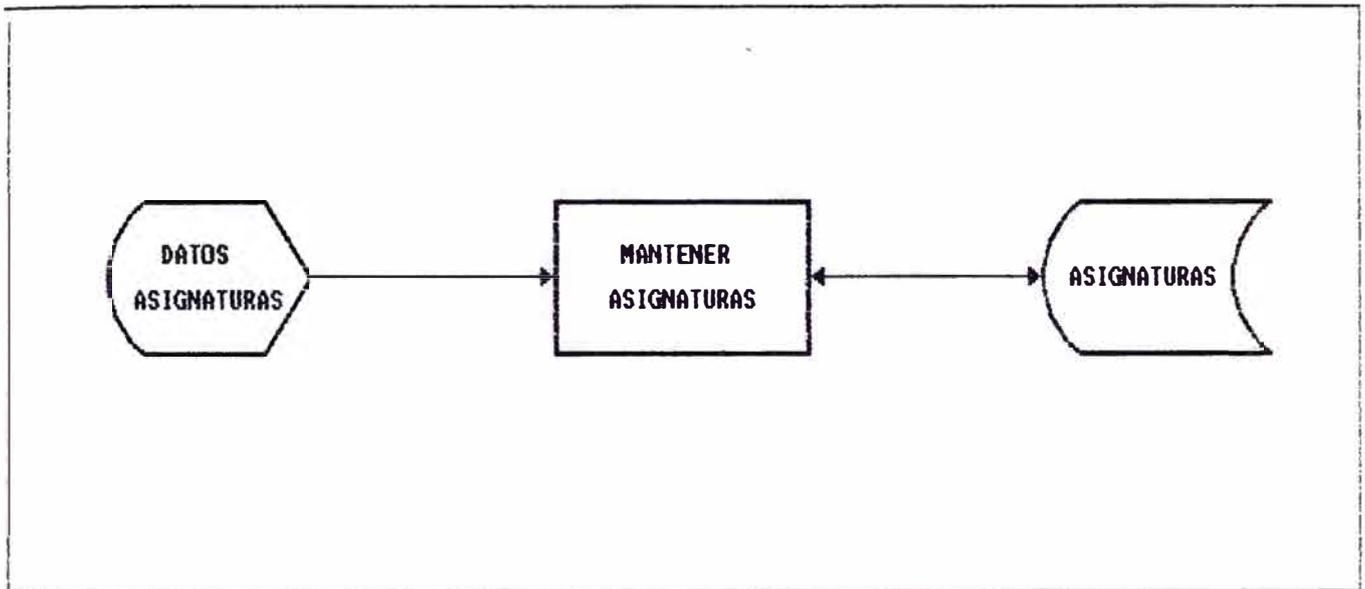
DIAGRAMA DEL PROCESO



REGLAS DEL PROCESO

```
INICIO
  ABRIR ARCHIVO PROFESORES
  INGRESE OPCION
  MIENTRAS OPCION <> FIN HACER
    SI OPCION = INGRESAR
      INGRESAR PROFESORES
    SI OPCION = ELIMINAR
      ELIMINAR PROFESORES
    SI OPCION = MODIFICAR
      MODIFICAR PROFESORES
  EN CASO CONTRARIO
    ERROR
  INGRESAR OPCION
  FIN-BUCLE
  CERRAR ARCHIVO PROFESORES
FIN
```

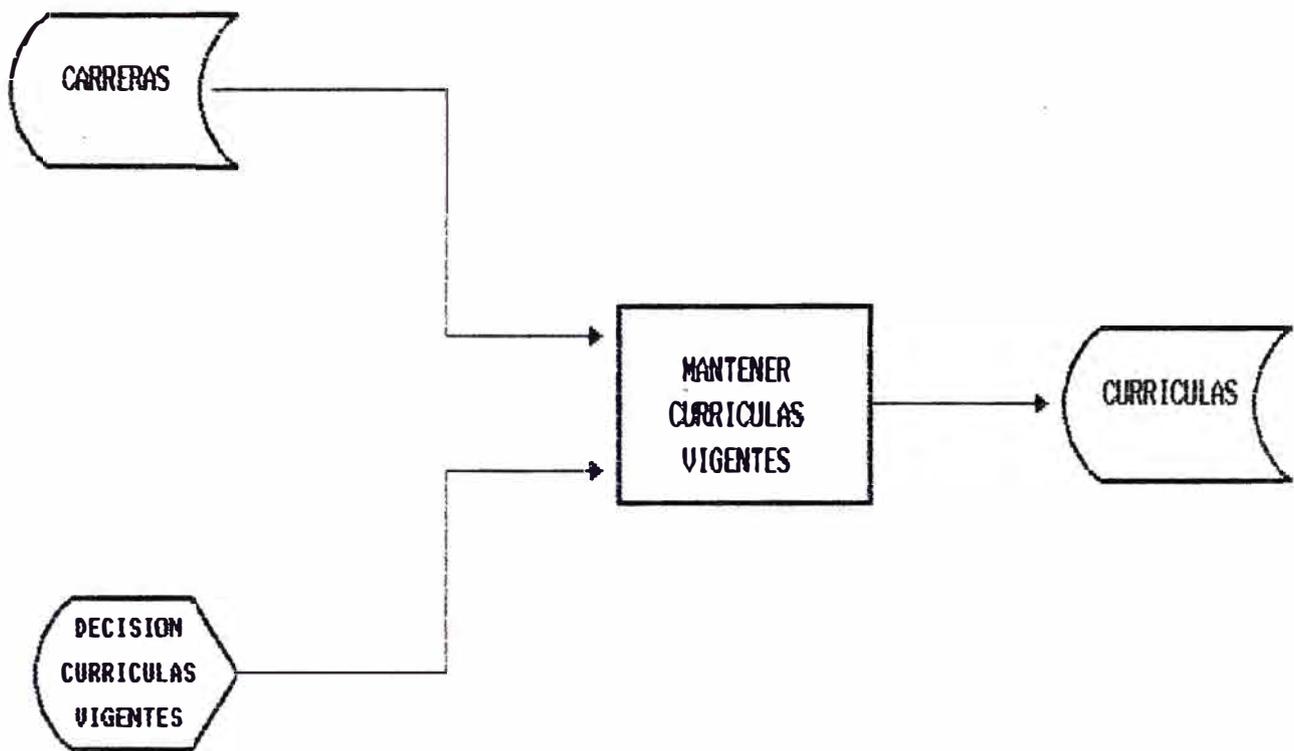
DIAGRAMA DEL PROCESO



REGLAS DEL PROCESO

```
INICIO
  ABRIR ARCHIVO ASIGNATURAS
  INGRESE OPCION
  MIENTRAS OPCION <> FIN HACER
    SI OPCION = INGRESAR
      INGRESAR ASIGNATURAS
    SI OPCION = ELIMINAR
      ELIMINAR ASIGNATURAS
    SI OPCION = MODIFICAR
      MODIFICAR ASIGNATURAS
  EN CASO CONTRARIO
    ERROR
  INGRESAR OPCION
FIN-BUCLE
CERRAR ARCHIVO ASIGNATURAS
FIN
```

DIAGRAMA DEL PROCESO



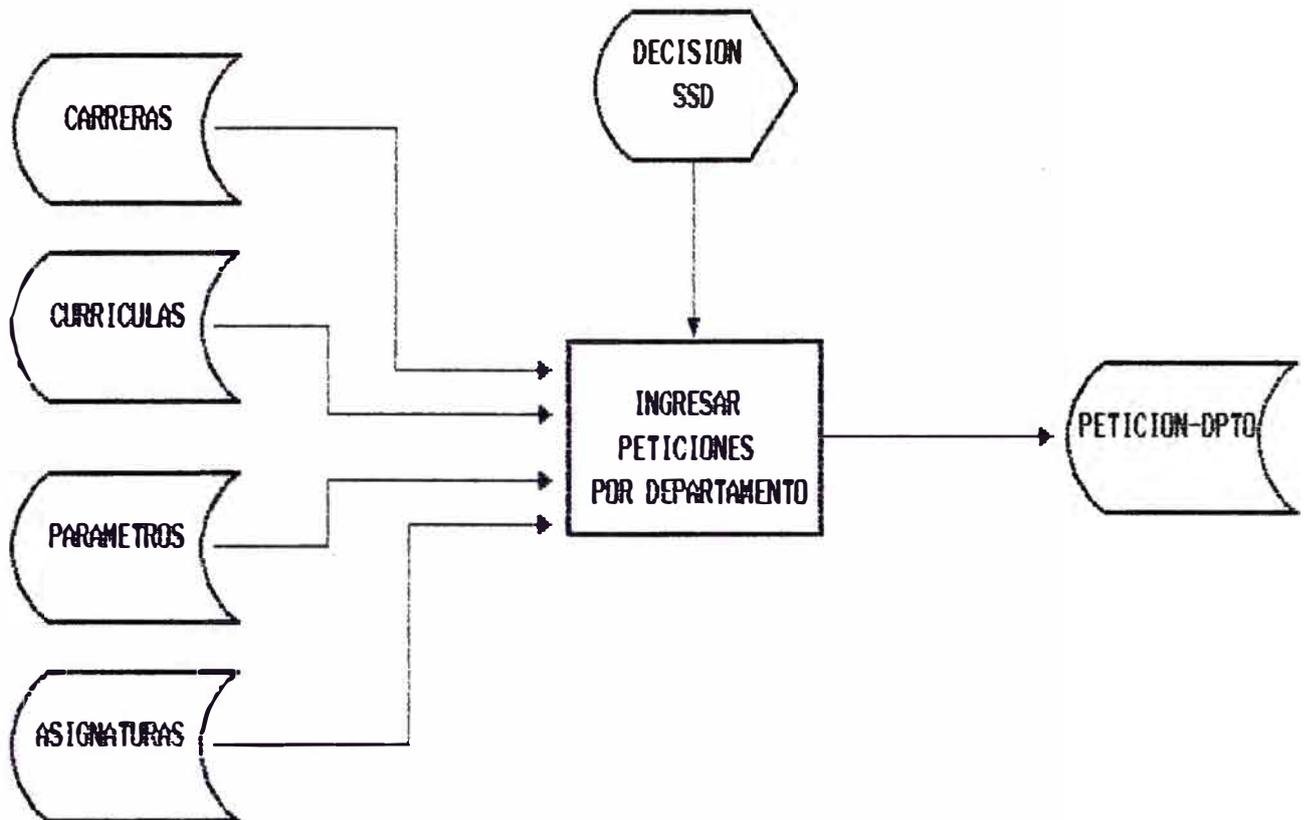
REGLAS DE PROCESO:

INICIO

ABRIR ARCHIVOS CARRERAS, CURRICULAS
INGRESAR CARRERA
BUSCAR CURRICULA
MIENTRAS NO EOF CURRICULA HACER
 INGRESAR DECISION
 ACTUALIZAR CURRICULA
 LLER CURRICULA
FIN-BUCLE
CERRAR ARCHIVOS CARRERAS, CURRICULAS

FIN

DIAGRAMA DEL PROCESO



* SSD = SOLICITUD DE SERVICIOS DOCENTES

REGLAS DE PROCESO:

INICIO

ABRIR ARCHIVOS CARRERAS, CURRICULAS, PARAMETROS, ASIGNATURAS, PETICION-DPTO

LEER REGISTRO PARAMETROS

LEER FECHAS ACTUAL

SI ESTA DENTRO DE LOS LIMITES INDICADOS POR LOS PARAMETROS

BUSCAR CARRERA PROCESADA EN CARRERAS

BUSCAR CURRICULA ASOCIADA A LA CARRERA

MIENTRAS HAYAN CURRICULAS VIGENTES HACER

LEER ASIGNATURAS ASOCIADAS A LA CURRICULA Y AL SEMESTRE EN CURSO

INGRESAR DECISION SSD

SI LA DECISION ES AFIRMATIVA

CARGAR PETICION-DPTO CON ASIGNATURAS DE LA CURRICULA CORRESPONDIENTE

AL SEMESTRE EN CURSO

FIN-CONDICION

BUSCAR SGTE. CURRICULA ASOCIADO A LA CARRERA

FIN-BUCLE

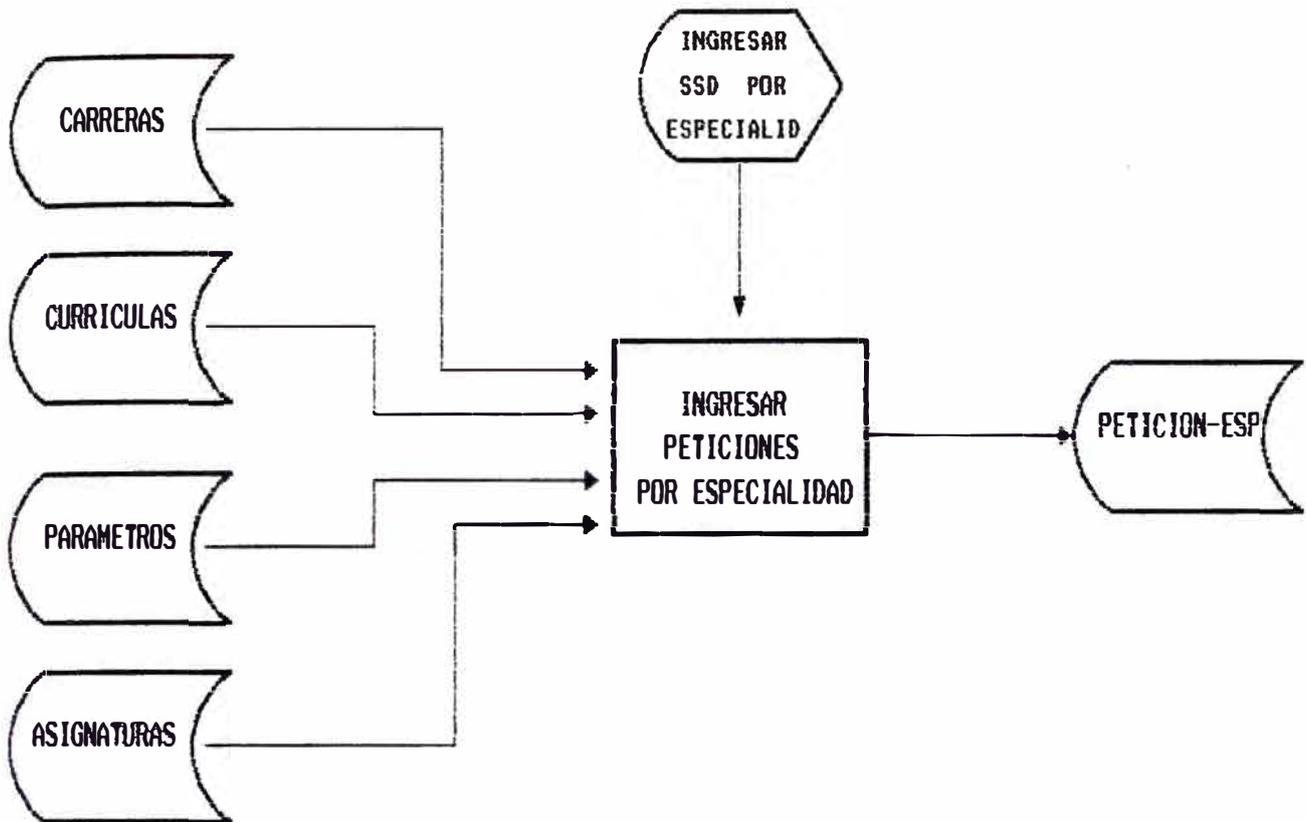
EN CASO CONTRARIO

ERROR OPERACIONAL

CERRAR ARCHIVOS

FIN

DIAGRAMA DEL PROCESO



REGLAS DE PROCESO:

INICIO

ABRIR ARCHIVOS CARRERAS, CURRICULAS, PARAMETROS, ASIGNATURAS, PETICION-ESP

LEER REGISTRO PARAMETROS

LEER FECHAS ACTUAL

SI ESTA DENTRO DE LOS LIMITES INDICADOS POR LOS PARAMETROS

BUSCAR CARRERA PROCESADA EN CARRERAS

BUSCAR CURRICULA ASOCIADA A LA CARRERA

MIENTRAS HAYAN CURRICULAS VIGENTES HACER

LEER ASIGNATURAS ASOCIADAS A LA CURRICULA Y AL SEMESTRE EN CURSO

INGRESAR SSD POR ESPECIALIDAD

CARGAR ARCHIVO PETICION-ESP CON LAS SSD POR ESPECIALIDAD

BUSCAR SIGUIENTE CURRICULA ASOCIADA A LA CARRERA

FIN-BUCLE

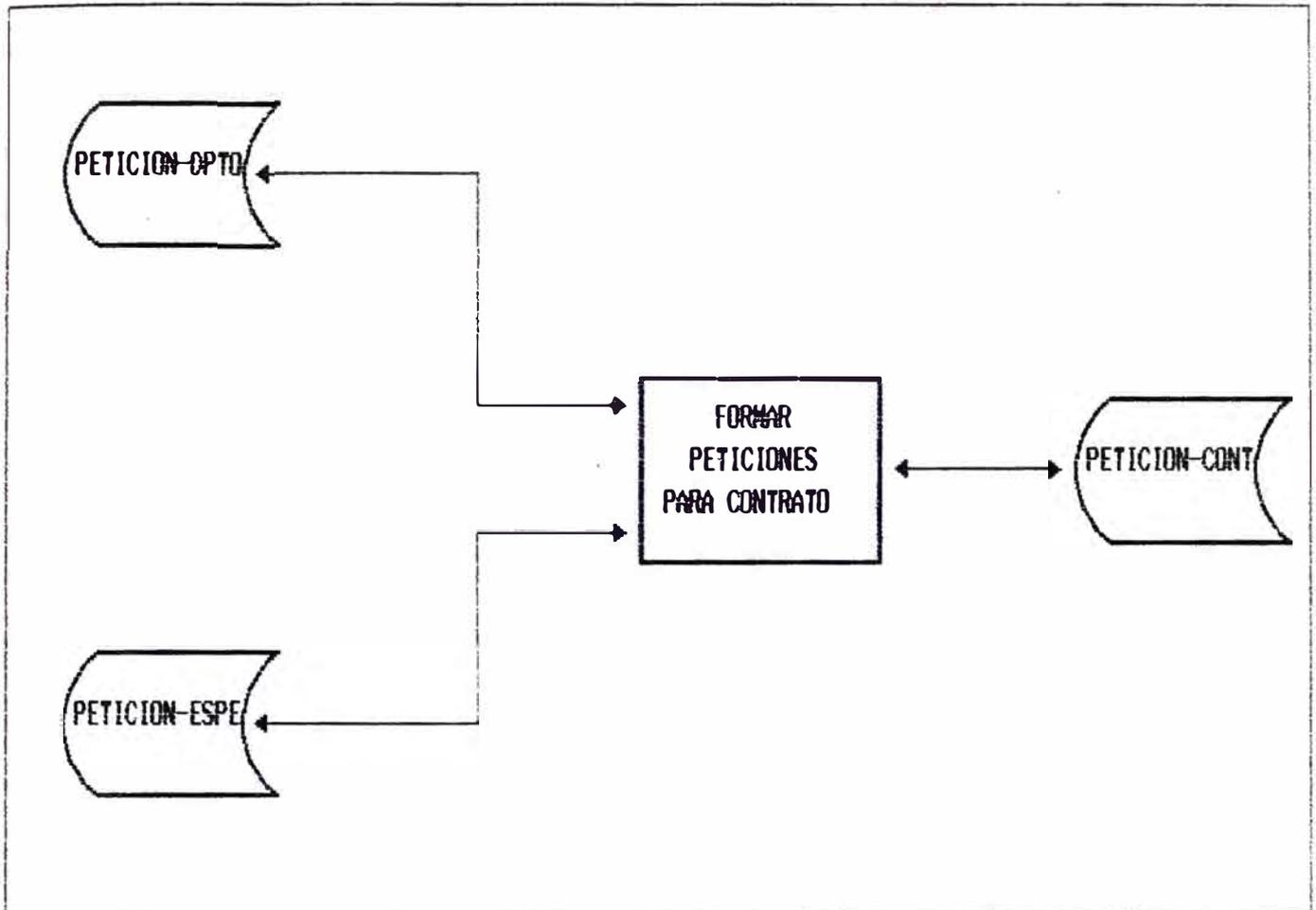
EN CASO CONTRARIO

ERROR OPERACIONAL

CERRAR ARCHIVOS

FIN

DIAGRAMA DEL PROCESO



REGLAS DE PROCESO:

INICIO

ABRIR ARCHIVOS PETICION-DPTO, PETICION-ESPE, PETICION-CONT

LEER ARCHIVOS PETICION-DPTO, PETICION-CONT

MIENTRAS NO EOF PETICION-DPTO O PETICION-CONT

SI NO MARCA-PROCESADO ENTONCES

CARGAR PETICION-CONT CON LOS DATOS PETICION-DPTO O PETICION-ESPE

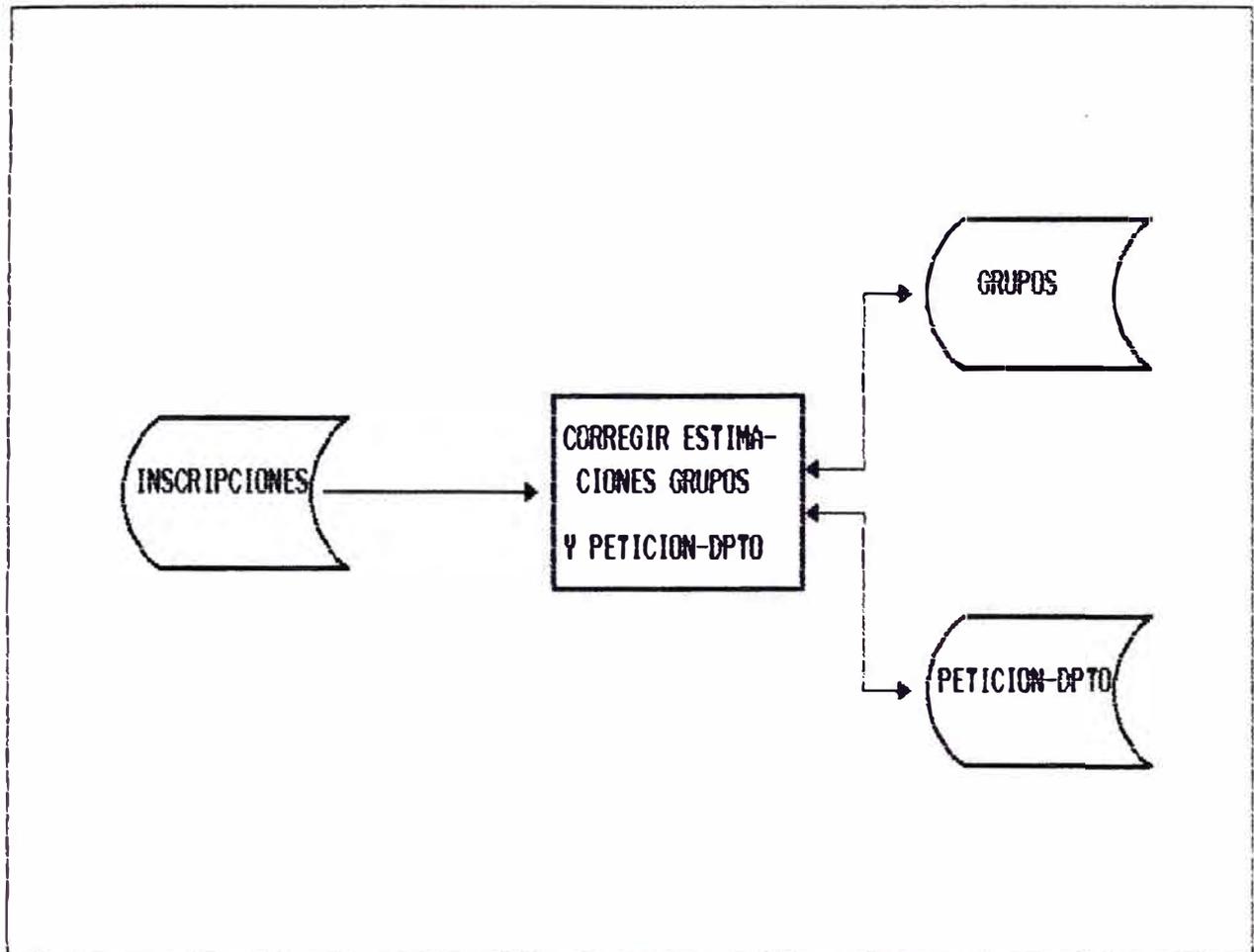
FIN-CONDICION

FIN-BUCLE

CERRAR ARCHIVOS PETICION-DPTO, PETICION-ESPE, PETICION-CONT

FIN

DIAGRAMA DEL PROCESO



REGLAS DE PROCESO:

INICIO

ABRIR ARCHIVOS INSCRIPCIONES, GRUPOS, PETICION-DPTO

LEER REGISTRO INSCRIPCIONES

MIENTRAS NOT EOF INSCRIPCIONES HACER

 BUSCAR ASIGNATURA EN EL ARCHIVO GRUPOS

 ACTUALIZAR DATOS DEL ARCHIVO GRUPOS

 BUSCAR ASIGNATURA EN EL ARCHIVO PETICION-DPTO

 ACTUALIZAR DATOS DEL ARCHIVO PETICION-DPTO

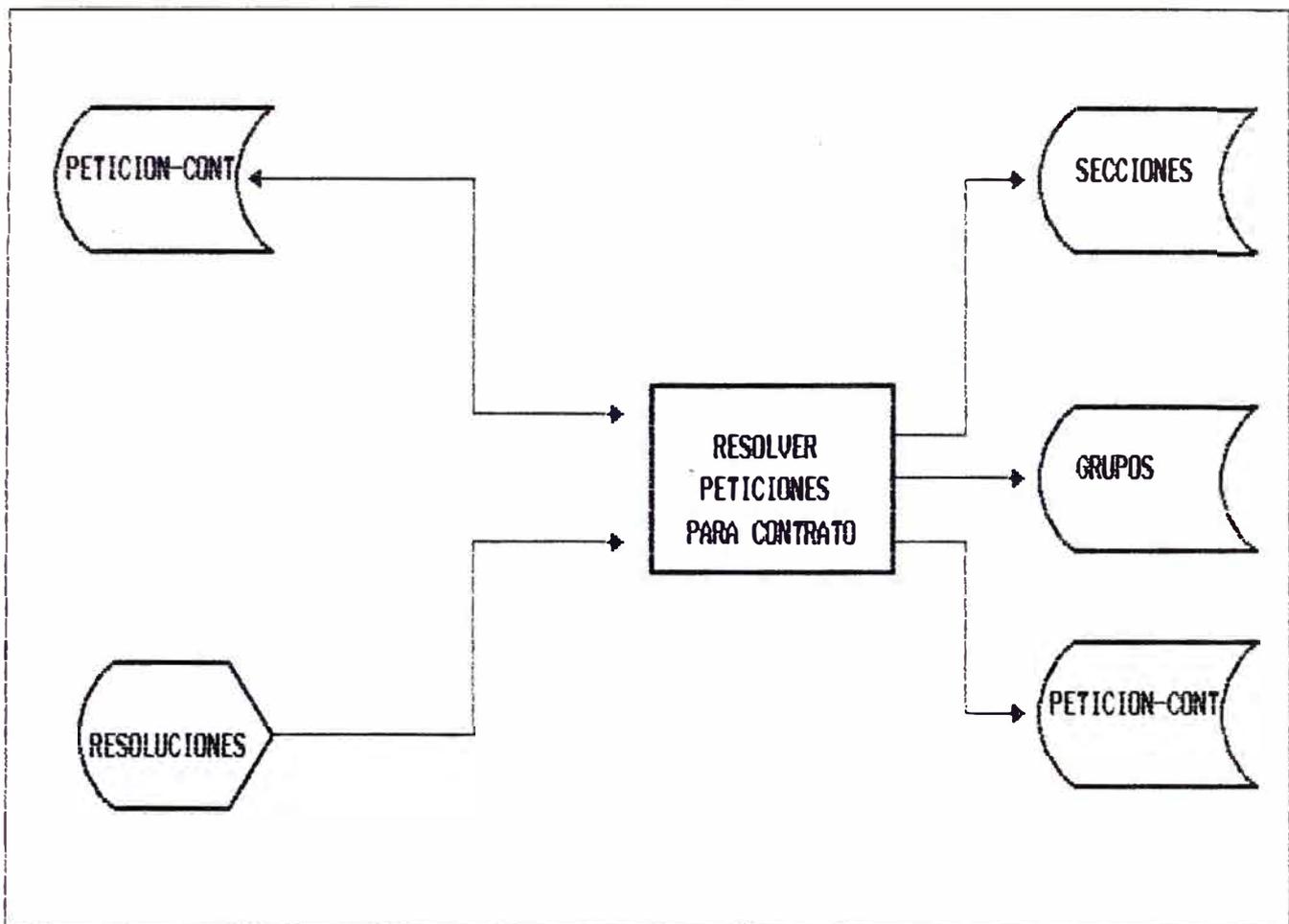
 LEER REGISTRO INSCRIPCIONES

FIN-BUCLE

CERRAR ARCHIVOS

FIN

DIAGRAMA DEL PROCESO



REGLAS DE PROCESO:

INICIO

ABRIR ARCHIVOS PETICION-CONT, SECCIONES, GRUPOS, DOCENCIA

LEER REGISTRO PETICION-CONT

MIENTRAS NO EOF PETICION-CONT HACER

 INGRESAR RESOLUCION

 SI RESOLUCION ES APROBADA

 CARGAR SECCIONES CON DATOS DE PETICION-CONT

 FORMAR GRUPOS

 CARGAR DOCENCIA CON DATOS PETICION-CONT

 FIN-CONDICION

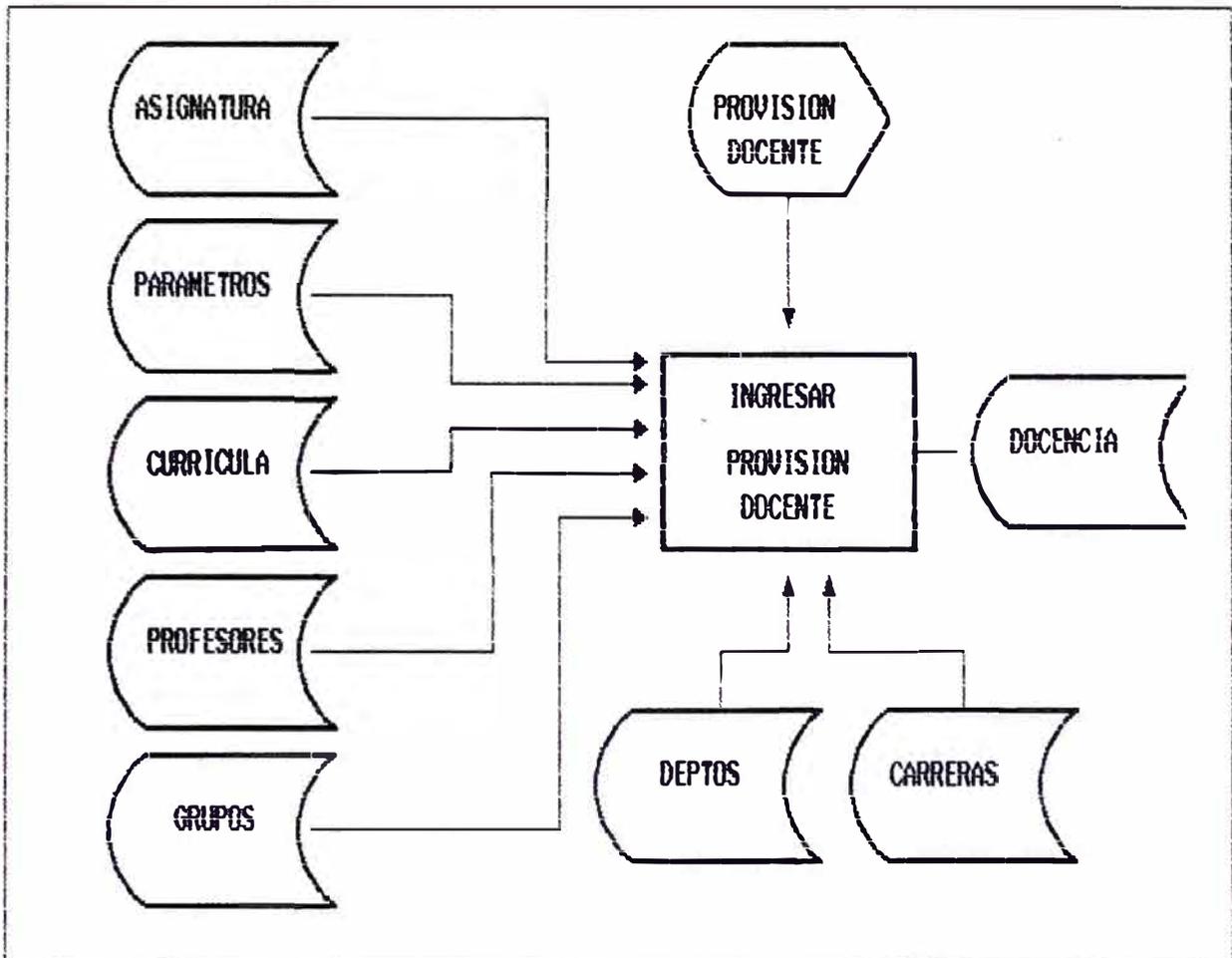
 LEER REGISTRO PETICION-CONT

FIN-BUCLE

CERRAR ARCHIVOS

FIN

DIAGRAMA DEL PROCESO



REGLAS DE PROCESO:

INICIO

ABRIR ARCHIVOS ASIGNATURAS, PARAMETROS, CURRICULA, PROFESORES,
DEPTOS, GRUPOS, CARRERAS, DOCENCIA

LEER REGISTRO PARAMETROS

LEER FECHA ACTUAL

SI ESTA DENTRO DE LOS LIMITES INDICADOS POR LOS PARAMETROS

INGRESAR LA PROVISION DOCENTE

ACTUALIZAR ARCHIVO DOCENCIA CON LA INFORMACION PROPORCIONADA

EN CASO CONTRARIO

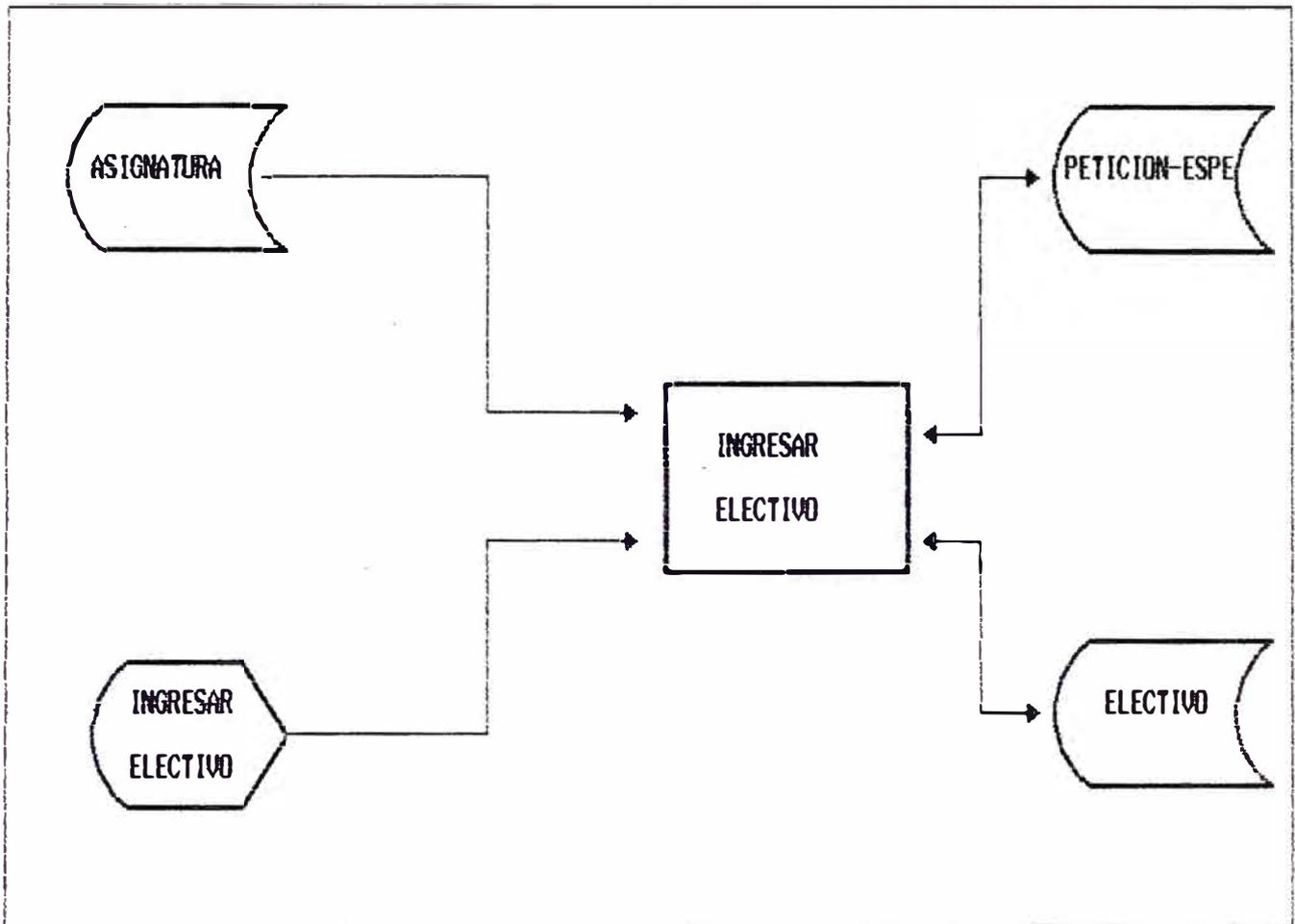
ERROR OPERACIONAL

FIN-CONDICION

CERRAR ARCHIVOS

FIN

DIAGRAMA DEL PROCESO



REGLAS DE PROCESO:

INICIO

ABRIR ARCHIVOS ASIGNATURAS, PETICION-ESPE, ELECTIVO

INGRESAR ELECTIVO

BUSCAR ELECTIVO EN ARCHIVO ASIGNATURA Y RESCATAR SUS DATOS

CARGAR LA INFORMACION CON RESPECTO A LA SSD DEL ELECTIVO EN EL ARCHIVO PETICION-ESPE

CARGAR DATOS RESCATADOS DEL ELECTIVO EN EL ARCHIVO ELECTIVO

CERRAR ARCHIVO

FIN

DIAGRAMA DEL PROCESO



REGLAS DEL PROCESO

```
INICIO
  ABRIR ARCHIVO PARAMETROS
  INGRESE OPCION
  MIENTRAS OPCION <> FIN HACER
    SI OPCION = INGRESAR
      INGRESAR PARAMETROS
    SI OPCION = ELIMINAR
      ELIMINAR PARAMETROS
    SI OPCION = MODIFICAR
      MODIFICAR PARAMETROS
  EN CASO CONTRARIO
    ERROR
  INGRESAR OPCION
  FIN-BUCLE
  CERRAR ARCHIVO PARAMETROS
FIN
```

| UNIVERSIDAD DE JULIACA CENTRO DE COMPUTO | | FECHA : | | PAG. No |
|---|--|--------------|-------|-----------------|
| DESCRIPCION DE ARCHIVOS | | | | |
| SISTEMA : SESTION DOCENTE | | SUBSISTEMA : | | |
| ARCHIVO : FACULTADES | DESCRIPCION : MANTIENE INFORMACION DE LAS FACULTADES EXISTENTES EN LA U.A.N.C.V. | | | |
| SOORTE : DISCO | | | | |
| ORGANIZACION : INDEXADA | | | | |
| CLAVE Nro. 1 : FAC-COD180 | | | | |
| CLAVE Nro. 2 : | | | | |
| CLAVE Nro. 3 : | | | | |
| CLAVE Nro. 4 : | FORMATO EN DISCO : FDR0011.FCD | | | |
| CLAVE Nro. 5 : | FORMATO REGISTOR : | | | |
| LARGO REGISTRO : | FACTOR BLOQUEO : | | | |
| MAX. REGISTROS : | LARGO BLOQUE : | | | |
| NOMBRE LOGICO : | | | | |
| DESCRIPCION DE REGISTRO | | | | |
| Pos. | NOMBRE | TIPO | LARGO | DESCRIPCION |
| 1 | FAC-COD180 | 9 | 1 | COD180 FACULTAD |
| 2 | FAC-NOMBRE | X | 38 | NOMBRE FACULTAD |

| UNIVERSIDAD DE JULIACA CENTRO DE COMPUTO | | FECHA : | | PAG. No |
|---|---|--------------|-------|---------------------|
| DESCRIPCION DE ARCHIVOS | | | | |
| SISTEMA : GESTION DOCENTE | | SUBSISTEMA : | | |
| ARCHIVO : DEPARTAMENTOS | DESCRIPCION : MANTIENE LOS DATOS DE LOS DEPARTAMENTOS EXISTENTES EN LA U.A.N.C.V. | | | |
| SOPORTE : DISCO | | | | |
| ORGANIZACION : INDEXADA | | | | |
| CLAVE Nro. 1 : DEP-CODIGO | | | | |
| CLAVE Nro. 2 : | | | | |
| CLAVE Nro. 3 : | | | | |
| CLAVE Nro. 4 : | | | | |
| CLAVE Nro. 5 : | FORMATO EN DISCO : FDAR0006.FCD | | | |
| LARGO REGISTRO : | FORMATO REGISTRO : | | | |
| MAX. REGISTROS : | FACTOR BLOQUEO : | | | |
| NOMBRE LOGICO : | LARGO BLOQUE : | | | |
| DESCRIPCION DE REGISTRO | | | | |
| Pos. | NOMBRE | TIPO | LARGO | DESCRIPCION |
| 1 | DEP-CODIGO | X | 2 | CODIGO DEPARTAMENTO |
| 3 | FAC-CODIGO | 9 | 1 | CODIGO FACULTAD |
| 4 | DEP-NOMBRE | X | 64 | NOMBRE DEPARTAMENTO |
| 68 | DEP-JEFE | X | 88 | JEFE DEPARTAMENTO |

| UNIVERSIDAD DE JULIACA CENTRO DE COMPUTO | | FECHA : | | PAG. No |
|---|---------------------|--|-------|--------------------------|
| DESCRIPCION DE ARCHIVOS | | | | |
| SISTEMA | : GESTION DOCENTE | SUBSISTEMA : | | |
| ARCHIVO | : CARRERAS | DESCRIPCION : MANTIENE LOS DATOS DE LAS CARRERAS EXISTENTES EN LA U.A.N.C.V. | | |
| SOPORTE | : DISCO | | | |
| ORGANIZACION | : INDEXADA | | | |
| CLAVE Nro. 1 | : CODIGO-CARRERA | | | |
| CLAVE Nro. 2 | : | | | |
| CLAVE Nro. 3 | : | | | |
| CLAVE Nro. 4 | : | FORMATO EN DISCO : FDAR0007.PCD | | |
| CLAVE Nro. 5 | : | FORMATO REGISTRO : | | |
| LARGO REGISTRO : | | FACTOR BLOQUEO : | | |
| MAX. REGISTROS : | | LARGO BLOQUE : | | |
| NOMBRE LOGICO : | | | | |
| DESCRIPCION DE REGISTRO | | | | |
| Pos. | NOMBRE | TIPO | LARGO | DESCRIPCION |
| 1 | CODIGO-CARRERA | | | |
| | FAC-CODIGO | 9 | 1 | CODIGO FACULTAD |
| | CAR-CORRELATIVO | 9 | 2 | CORRELATIVO CARRERA |
| 4 | CAR-NOMBRE | X | 40 | NOMBRE CARRERA ABREVIADO |
| 44 | CAR-NOMBRE-COMPLETO | X | 64 | NOMBRE CARRERA COMPLETO |
| 108 | CAR-DOLAR | 9(8)V99 | 10 | VALOR CARRERA DOLAR |
| 118 | CAR-SOLES | 9(8)V99 | 10 | VALOR CARRERA SOLES |
| 128 | CAR-CURRI-ULTIMO | 9 | 2 | ULTIMA CURRICULA CARRERA |

| UNIVERSIDAD DE JULIACA CENTRO DE COMPUTO | | FECHA : | FAB. No | |
|---|-----------------------|--|---------|---------------------------------|
| DESCRIPCION DE ARCHIVOS | | | | |
| SISTEMA : GESTION DOCENTE | | SUBSISTEMA : | | |
| ARCHIVO : GRUPOS | | DESCRIPCION : ESTE ARCHIVO MANTIENE TODO LOS GRUPOS DE ASIGNACION QUE SE FORMARON A PARTIR DE LAS PETICIONES DE LOS DIRECTORES DE CARRERA ADEMÁS LOS CURSOS QUE NECESITA SECRETARIA ACADEMICA PARA LAS INSCRIPCIONES. CON LA PROVISION DOCENTE SUMINISTRADA POR LOS JEFES DE DEPARTAMENTO LOS GRUPOS PROPORCIONAN LA BASE PARA OBTENER LA CARGA DOCENTE DE CADA ACADEMICO Y DEL DEPARTAMENTO. | | |
| SOPORTE : DISCO | | | | |
| ORGANIZACION : INDEXADA | | | | |
| CLAVE Nro. 1 : CODIGO-GRUPO | | | | |
| CLAVE Nro. 2 : | | | | |
| CLAVE Nro. 3 : | | | | |
| CLAVE Nro. 4 : | | | | |
| CLAVE Nro. 5 : | | | | |
| LARGO REGISTRO : | | | | FORMATO EN DISCO : F0AR0004.PCD |
| MAX. REGISTROS : | | FORMATO REGISTRO : | | |
| NOMBRE LOGICO : | | FACTOR BLOQUEO : | | |
| | | LARGO BLOQUE : | | |
| DESCRIPCION DE REGISTRO | | | | |
| POS. | NOMBRE | TIPO | LARGO | DESCRIPCION |
| 1 | CODIGO GRUPO | | | |
| | SEC-CLAVE | X | 1 | CLAVE SECCION |
| | GRU-CLAVE | X | 1 | CLAVE GRUPO |
| | DEP-CODIGO | X | 2 | CODIGO DEPARTAMENTO |
| | ASI-CORRELATIVO | ? | 3 | CORRELATIVO ASIGNATURA |
| 8 | GRU-NRO-ALU-EST | ? | 3 | ALUMNOS ESTIMADOS |
| 11 | GRU-NRO-ALU-REA | ? | 3 | ALUMNOS REALES |
| 14 | GRU-NRO-ALU-HORAS | ? | 2 | TOTAL HORAS |
| 16 | GRU-NRO-HRS-CUBIERTAS | ? | 2 | HORAS CUBIERTAS |
| 18 | GRU-BALA-CURSO | X | 3 | BALA CURSO |

| UNIVERSIDAD DE JULIACA CENTRO DE COMPUTO | | FECHA : | PAG. No | |
|---|--|--------------|---------|---------------------------------|
| DESCRIPCION DE ARCHIVOS | | | | |
| SISTEMA : GESTION DOCENTE | | SUBSISTEMA : | | |
| ARCHIVO : GRUPOS | DESCRIPCION : ESTE ARCHIVO MANTIENE TODO LOS GRUPOS DE ASIGNACION QUE SE FORMARON A PARTIR DE LAS PETICIONES DE LOS DIRECTORES DE CARRERA ADEMAS LOS CURSOS QUE NECESITA SECRETARIA ACADEMICA PARA LAS INSCRIPCIONES. CON LA PROVISION DOCENTE SUMINISTRADA POR LOS JEFES DE DEPARTAMENTO LOS GRUPOS PROPORCIONAN LA BASE PARA OBTENER LA CARSA DOCENTE DE CADA ACADEMICO Y DEL DEPARTAMENTO. | | | |
| SOPORTE : DISCO | | | | |
| ORGANIZACION : INDEXADA | | | | |
| CLAVE Nro. 1 : CODIGO-GRUPO | | | | |
| CLAVE Nro. 2 : | | | | |
| CLAVE Nro. 3 : | | | | |
| CLAVE Nro. 4 : | | | | |
| CLAVE Nro. 5 : | | | | |
| LARGO REGISTRO : | | | | FORMATO EN DISCO : F0AR0004.FCD |
| MAX. REGISTROS : | FORMATO REGISTOR : | | | |
| NOMBRE LOGICO : | FACTOR BLOQUEO : | | | |
| | LARGO BLOQUE : | | | |
| DESCRIPCION DE REGISTRO | | | | |
| Pos. | NOMBRE | TIPO | LARGO | DESCRIPCION |
| 1 | CODIGO GRUPO | | | |
| | SEC-CLAVE | X | 1 | CLAVE SECCION |
| | GRU-CLAVE | X | 1 | CLAVE GRUPO |
| | DEP-CODIGO | X | 2 | CODIGO DEPARTAMENTO |
| | ABI-CORRELATIVO | 9 | 3 | CORRELATIVO ASIGNATURA |
| 8 | GRU-NRO-ALU-EST | 9 | 3 | ALUMNOS ESTIMADOS |
| 11 | GRU-NRO-ALU-REA | 9 | 3 | ALUMNOS REALES |
| 14 | GRU-NRO-ALU-HORAS | 9 | 2 | TOTAL HORAS |
| 16 | GRU-NRO-HRS-CUBIERTAS | 9 | 2 | HORAS CUBIERTAS |
| 18 | GRU-SALA-CURSO | X | 3 | SALA CURSO |

| UNIVERSIDAD DE JULIACA CENTRO DE COMPUTO | | FECHA : | | PAG. No |
|---|------------------|---|-------|------------------------|
| DESCRIPCION DE ARCHIVOS | | | | |
| SISTEMA : SECCION DOCENTE | | SUBSISTEMA : | | |
| ARCHIVO : SECCIONES | | DESCRIPCION : RECOSE LAS SECCIONES DE CARRERAS QUE SE ORIGINAN DE LAS PETICIONES POR DEPARTAMENTO. | | |
| SOPORTE : DISCO | | | | |
| ORGANIZACION : INDEXADA | | | | |
| CLAVE Nro. 1 : COD180-SECCIONES | | | | |
| CLAVE Nro. 2 : | | | | |
| CLAVE Nro. 3 : | | | | |
| CLAVE Nro. 4 : | | FORMATO EN DISCO : FDAR0015.FCD | | |
| CLAVE Nro. 5 : | | FORMATO REGISTRO : | | |
| LARGO REGISTRO : | | FACTOR BLOQUEO : | | |
| MAX. REGISTROS : | | LARGO BLOQUE : | | |
| NOMBRE LOGICO : | | | | |
| DESCRIPCION DE REGISTRO | | | | |
| Pos. | NOMBRE | TIPO | LARGO | DESCRIPCION |
| 1 | COD180-SECCIONES | | | |
| | SEC-CLAVE | X | 1 | CLAVE SECCION |
| | DEP-COD180 | X | 2 | COD180 DEPARTAMENTO |
| | ASI-CORRELATIVO | 9 | 3 | CORRELATIVO ASIGNATURA |
| 7 | SEC-NRO-ALUMNOS | 9 | 3 | NUMERO DE ALUMNOS |
| 10 | SEC-GRUPOS-CAT | 9 | 2 | GRUPOS DE CATEDRA |
| 12 | SEC-GRUPOS-PRA | 9 | 2 | GRUPOS DE PRACTICA |
| 14 | SEC-GRUPOS-LAB | 9 | 2 | GRUPOS DE LABORATORIO |
| 16 | SEC-HORAS-CAT | 9 | 2 | HORAS DE CATEDRA |
| 18 | SEC-HORAS-PRA | 9 | 2 | HORAS DE PRACTICA |
| 20 | SEC-HORAS-LAB | 9 | 2 | HORAS DE LABORATORIO |

| UNIVERSIDAD DE JULIACA CENTRO DE COMPUTO | | FECHA : | PAG. No | |
|---|----------------------|--|---------|-------------------------------|
| DESCRIPCION DE ARCHIVOS | | | | |
| SISTEMA : GESTION DOCENTE | | SUBSISTEMA : | | |
| ARCHIVO : PROFESORES | | DESCRIPCION : MANTIENE INFORMACION DE LOS PROFESORES ORDINARIOS Y CONTRATADOS. | | |
| SOPORTE : DISCO | | | | |
| ORGANIZACION : INDEXADA | | | | |
| CLAVE Nro. 1 : COD100-PROFESOR | | | | |
| CLAVE Nro. 2 : | | | | |
| CLAVE Nro. 3 : | | | | |
| CLAVE Nro. 4 : | | FORMATO EN DISCO : FDAR0009.FCD | | |
| CLAVE Nro. 5 : | | FORMATO REGISTRO : | | |
| LARGO REGISTRO : | | FACTOR BLOQUEO : | | |
| MAX. REGISTROS : | | LARGO BLOQUE : | | |
| NOMBRE LOGICO : | | | | |
| DESCRIPCION DE REGISTRO | | | | |
| Pas. | NOMBRE | TIPO | LARGO | DESCRIPCION |
| 1 | COD100-PROFESOR | | | |
| | AF-NRO-LE | ? | 8 | NUMERO LE |
| | AF-DIS-LE | X | 1 | DIGITO VERIFICADOR LE |
| 10 | DEF-COD100 | X | 2 | COD100 DEPARTAMENTO |
| 12 | PRO-NOMBRE | X | 20 | NOMBRE PROFESOR |
| 32 | PRO-PATERNO | X | 19 | APELLIDO PATERNO PROFESOR |
| 47 | PRO-MATERNO | X | 19 | APELLIDO MATERNO PROFESOR |
| 62 | PRO-CONTRATO | ? | 2 | TIPO CONTRATO PROFESOR |
| 64 | PRO-DEDICACION | ? | 1 | DEDICACION PROFESOR |
| 65 | PRO-NSRA-VIS | ? | 2 | NUMERO GRADO VIBENTE PROFESOR |
| 67 | PRO-CGRA-VIS | X | 1 | CATEGORIA GRADO VIBENTE PROF. |
| 68 | PRO-SITUACION | ? | 1 | SITUACION PROFESOR |
| 69 | PRO-NRO-HRS-CONTRATO | ? | 2 | NUMERO DE HORAS CONTRATO |
| 71 | PRO-NRO-QUINGUENIO | ? | 2 | NUMERO DE QUINGUENIOS |
| 73 | PRO-FEC-INS-INS | DATE | 6 | FECHA INGRESO INSTITUCION |
| 79 | PRO-FEC-EGR-INS | DATE | 6 | FECHA EGRESO INSTITUCION |

| UNIVERSIDAD DE JULIACA CENTRO DE COMPUTO | | FECHA : | | PAG. No |
|---|-----------------|---|-------|-----------------------------|
| DESCRIPCION DE ARCHIVOS | | | | |
| SISTEMA : GESTION DOCENTE | | SUBSISTEMA : | | |
| ARCHIVO : DOCENCIA | | DESCRIPCION : MANTIENE INFORMACION SOBRE LOS GRUPOS FORMADOS EN EL SEMESTRE EN CURSO DE CADA ASIGNATURA JUNTO CON LOS PROFESORES INCORPORADOS EN CADA GRUPO | | |
| SOPORTE : DISCO | | | | |
| ORGANIZACION : INDEXADA | | | | |
| CLAVE Nro. 1 : CODIGO-DOCENCIA | | | | |
| CLAVE Nro. 2 : | | | | |
| CLAVE Nro. 3 : | | | | |
| CLAVE Nro. 4 : | | | | |
| CLAVE Nro. 5 : | | FORMATO EN DISCO : FDAR0010.FCD | | |
| LARGO REGISTRO : | | FORMATO REGISTRO : | | |
| MAX. REGISTROS : | | FACTOR BLOQUEO : | | |
| NOMBRE LOGICO : | | LARGO BLOQUE : | | |
| DESCRIPCION DE REGISTRO | | | | |
| Pos. | NOMBRE | TIPO | LARGO | DESCRIPCION |
| 1 | CODIGO-DOCENCIA | | | |
| | ABR-CLAVE | X | 1 | CLAVE ABRUPACION |
| | GRU-CLAVE | X | 1 | CLAVE GRUPO |
| | GRU-TIPO | X | 1 | TIPO ASIGNATURA |
| | DEP-CODIGO | X | 2 | CODIGO DEPARTAMENTO |
| | ABI-CORRELATIVO | 9 | 3 | CORRELATIVO ASIGNATURA |
| 17 | AF-NRO-LE | 9 | 8 | NUMERO LE PROFESOR |
| | AF-DIS-LE | X | 1 | DIGITO VERIFICADOR LE PROF. |
| | DOC-NRO-HORAS | 9 | 2 | NUMERO DE HORAS DE DOCENCIA |

| UNIVERSIDAD DE JULIACA CENTRO DE COMPUTO | | FECHA : | | PAG. No |
|---|---------------------|---------|---|------------------------------|
| DESCRIPCION DE ARCHIVOS | | | | |
| SISTEMA : GESTION DOCENTE | | | SUBSISTEMA : | |
| ARCHIVO : ASIGNATURAS | | | DESCRIPCION : MANTIENE INFORMACION DE LAS ASIGNATURAS IMPARTIDAS POR LOS DIFERENTES DEPARTAMENTOS DE LA U.A.N.C.V. TAMBIEN SE INCLUYEN AQUELLAS ASIGNATURAS DE ESPECIALIDAD. | |
| SOPORTE : DISCO | | | | |
| ORGANIZACION : INDEXADA | | | | |
| CLAVE Nro. 1 : CODIGO-ASIGNATURA | | | | |
| CLAVE Nro. 2 : | | | | |
| CLAVE Nro. 3 : | | | | |
| CLAVE Nro. 4 : | | | | |
| CLAVE Nro. 5 : | | | FORMATO EN DISCO : FDAR0003.FCD | |
| LARGO REGISTRO : | | | FORMATO REGISTRO : | |
| MAX. REGISTROS : | | | FACTOR BLOQUEO : | |
| NOMBRE LOGICO : | | | LARGO BLOQUE : | |
| DESCRIPCION DE REGISTRO | | | | |
| Pos. | NOMBRE | TIPO | LARGO | DESCRIPCION |
| 1 | CODIGO-ASIGNATURA | | | |
| | DEP-CODIGO | X | 2 | CODIGO DEPARTAMENTO |
| | ASI-CORRELATIVO | Y | 3 | CORRELATIVO ASIGNATURA |
| 6 | ASI-NOMBRE | X | 40 | NOMBRE ABREVIADO DE ASIGNAT. |
| 46 | ASI-NOMBRE-COMPLETO | X | 64 | NOMBRE COMPLETO ASIGNATURA |

| UNIVERSIDAD DE JULIACA CENTRO DE COMPUTO | | FECHA : | PAG. No | |
|---|-----------------------|---|---------|------------------|
| DESCRIPCION DE ARCHIVOS | | | | |
| SISTEMA : GESTION DOCENTE | | SUBSISTEMA : | | |
| ARCHIVO : CURRICULA | | DESCRIPCION : RECOBE LAS CURRICULAS DE ESTUDIOS DE LAS DIFERENTES CARRERAS DE LA U.A.N.C.V. | | |
| SOPORTE : DISCO | | | | |
| ORGANIZACION : INDEXADA | | | | |
| CLAVE Nro. 1 : CODIGO-CURRICULA-ESTUDIO | | | | |
| CLAVE Nro. 2 : | | | | |
| CLAVE Nro. 3 : | | FORMATO EN DISCO : FDAR0012.FCD | | |
| CLAVE Nro. 4 : | | | | |
| CLAVE Nro. 5 : | | | | |
| LARGO REGISTRO : | | FORMATO REGISTRO : | | |
| MAX. REGISTROS : | | FACTOR BLOQUEO : | | |
| NOMBRE LOGICO : | | LARGO BLOQUE : | | |
| DESCRIPCION DE REGISTRO | | | | |
| Pos. | NOMBRE | TIPO | LARGO | DESCRIPCION |
| 1 | COD-CURRICULA-ESTUDIO | 9 | 1 | COD.CURR.ESTUDIO |
| | FAC-CODIGO | 9 | 2 | CODIGO FACULTAD |
| | CAR-CORRELATIVO | 9 | 2 | CORRELA. CARRERA |
| | CUR-ANO | 9 | 2 | ANO CURRICULA |
| 6 | CUR-NRO-RESOLUCION | X | 5 | NRO RESOLUCION |
| 11 | CUR-FECHA-RESOLUCION | DATE | 6 | FECHA RESOLUCION |
| 17 | CUR-DURACION-SEMESTRE | 9 | 2 | DURACION SEMEST. |
| 19 | CUR-SEM-INICIO-VIG | 9 | 2 | SEM.INI.VIGENCIA |
| 21 | CUR-SEM-FIN-VIG | 9 | 2 | SEM.FIN VIGENCIA |

| UNIVERSIDAD DE JULIACA CENTRO DE COMPUTO | | FECHA : | PAG. No | |
|---|--------------------|--------------------|--|-------------------------------|
| DESCRIPCION DE ARCHIVOS | | | | |
| SISTEMA : | SESTION DOCENTE | SUBSISTEMA : | | |
| ARCHIVO : | INSCRIPCIONES | DESCRIPCION : | MANTIENE INFORMACION DE LOS ALUMNOS INSCRITOS EN UN DETERMINADO CURSO DURANTE EL SEMESTRE. | |
| SOPORTE : | DISCO | | | |
| ORGANIZACION : | INDEXADA | | | |
| CLAVE Nro. 1 : | COD180-INSCRIPCION | | | |
| CLAVE Nro. 2 : | | | | |
| CLAVE Nro. 3 : | | | | |
| CLAVE Nro. 4 : | | FORMATO EN DISCO : | FDAR0014.FCD | |
| CLAVE Nro. 5 : | | FORMATO REGISTOR : | | |
| LARGO REGISTRO : | | FACTOR BLOQUEO : | | |
| MAX. REGISTROS : | | LARGO BLOQUE : | | |
| NOMBRE LOGICO : | | | | |
| DESCRIPCION DE REGISTRO | | | | |
| Pos. | NOMBRE | TIPO | LARGO | DESCRIPCION |
| 1 | COD180-INSCRIPCION | 9 | 8 | NUMERO MATRICULA ALUMNO |
| | INS-NUM-MAT-ALU | X | 1 | DIGITO VERIFICADOR DE NUM.MAT |
| | INS-DIG-MAT-ALU | 9 | 1 | COD180 FACULTAD |
| | FAC-COD180 | 9 | 2 | CORRELATIVO CARRERA |
| | CAR-CORRELATIVO | X | 2 | COD180 DEPARTAMENTO |
| | DEP-COD180 | 9 | 3 | CORRELATIVO ASIGNATURA |
| | ASI-CORRELATIVO | 9 | 1 | CONDICION INSCRIPCION |
| 18 | INS-CONDICION | 9 | 2 | SEMESTRE INSCRIPCION |
| 19 | INS-SEMESTRE | 9 | 2 | ANO DE INGRESO |
| 21 | INS-ANO-INGRESO | 9 | 1 | RATIFICACION INSCRIPCION |
| 23 | INS-RATIFICACION | DATE | 6 | FECHA INSCRIPCION |
| 24 | INS-FECHA | X | 10 | PERSONA QUE INGRESO LA INSC. |
| 30 | INS-USUARIO | 9 | 2 | ANO DE LA CURRICULA |
| 45 | CUR-ANO | | | |

| UNIVERSIDAD DE JULIACA | | FECHA : | | PAG. No |
|-------------------------------------|--------------------|--|-------|------------------|
| CENTRO DE COMPUTO | | DESCRIPCION DE ARCHIVOS | | |
| SISTEMA : GESTION DOCENTE | | SUBSISTEMA : | | |
| ARCHIVO : PETICION-CONT | | DESCRIPCION : ESTE ARCHIVO MANTIENE TODAS LAS ASIGNATURAS SOLICITADAS POR LOS DIRECTORES DE CARRERAS QUE SEAN PARA CONTRATO. | | |
| SOPORTE : DISCO | | | | |
| ORGANIZACION : INDEXADA | | | | |
| CLAVE Nro. 1 : CODIGO-PETICION-CONT | | | | |
| CLAVE Nro. 2 : | | | | |
| CLAVE Nro. 3 : | | FORMATO EN DISCO : FDAR0002.FCD | | |
| CLAVE Nro. 4 : | | | | |
| CLAVE Nro. 5 : | | FORMATO REGISTRO : | | |
| LARGO REGISTRO : | | FACTOR BLOQUEO : | | |
| MAX. REGISTROS : | | LARGO BLOQUE : | | |
| NOMBRE LOGICO : | | | | |
| DESCRIPCION DE REGISTRO | | | | |
| Pos. | NOMBRE | TIPO | LARGO | DESCRIPCION |
| 1 | COD-PETICION-CONT | | | |
| | FAC-CODIGO | 9 | 1 | CODIGO FACULTAD |
| | CAR-CORRELATIVO | 9 | 2 | CORRELATIVO CARR |
| | CUR-ANO | 9 | 2 | ANO CURRICULA |
| | DEP-CODIGO | X | 2 | CODIGO DEPARTAM |
| 11 | ASI-CORRELATIVO | 9 | 3 | CORRELAT ASIGNAT |
| | FEL-MARC-PROCESADA | X | 1 | MARCA FETIC PROC |
| 12 | FEL-FECHA-INGRESO | DATE | 6 | FECHA INGR FETIC |

| UNIVERSIDAD DE JULIACA CENTRO DE COMPUTO | | FECHA : | | PAG. No |
|---|-------------------|---------|---|-------------------------------|
| DESCRIPCION DE ARCHIVOS | | | | |
| SISTEMA : GESTION DOCENTE | | | SUBSISTEMA : | |
| ARCHIVO : PARAMETROS | | | DESCRIPCION : MANTIENE INFORMACION SOBRE LOS PARAMETROS DEL SISTEMA ES DECIR, SEMESTRE, FECHA ENTREGA DE SSD Y PLAZO DE PROVICION DOCENTE. | |
| SOPORTE : DISCO | | | | |
| ORGANIZACION : INDEXADA | | | | |
| CLAVE Nro. 1 : CODIGO-PARAMETRO | | | | |
| CLAVE Nro. 2 : | | | | |
| CLAVE Nro. 3 : | | | | |
| CLAVE Nro. 4 : | | | | |
| CLAVE Nro. 5 : | | | FORMATO EN DISCO : F0AR0009.FCD | |
| LARGO REGISTRO : | | | FORMATO REGISTOR : | |
| MAX. REGISTROS : | | | FACTOR BLOQUEO : | |
| NOMBRE LOGICO : | | | LARGO BLOQUE : | |
| DESCRIPCION DE REGISTRO | | | | |
| Pos. | NOMBRE | TIPO | LARGO | DESCRIPCION |
| 1 | PAR-CODIGO | ? | 1 | CODIGO PARAMETRO |
| 2 | PAR-PLAZO-SSD | DATE | 4 | FECHA DE SOLIC. SERV. DOCENT. |
| 8 | PAR-PLAZO-ASI-DOC | DATE | 6 | FECHA DE ASIGNACION DOCENTE |
| 14 | PAR-ANO-SEMESTRE | ? | 2 | ANO DEL SEMESTRE EN CURSO |
| 16 | PAR-SEMESTRE | ? | 1 | SEMESTRE EN CURSO |

| UNIVERSIDAD DE JULIACA CENTRO DE COMPUTO | | FECHA : | PAG. No | |
|---|----------------------|--------------------|---|-------------------------------|
| DESCRIPCION DE ARCHIVOS | | | | |
| SISTEMA : | GESTION DOCENTE | SUBSISTEMA : | | |
| ARCHIVO : | PETICION-DPTO | DESCRIPCION : | MANTIENE INFORMACION SOBRE LAS SOLICITUDES DE SERVICIOS DOCENTES PROPORCIONADOS POR CADA UN DE LOS JEFES DE DEPARTAMENTO. | |
| SOPORTE : | DISCO | | | |
| ORGANIZACION : | INDEXADA | | | |
| CLAVE Nro. 1 : | CODIGO-PETICION-DPTO | | | |
| CLAVE Nro. 2 : | | | | |
| CLAVE Nro. 3 : | | | | |
| CLAVE Nro. 4 : | | | | |
| CLAVE Nro. 5 : | | FORMATO EN DISCO : | FDAR0001.FCD | |
| LARGO REGISTRO : | | FORMATO REGISTOR : | | |
| MAX. REGISTROS : | | FACTOR BLOQUEO : | | |
| NOMBRE LOGICO : | | LARGO BLOQUE : | | |
| DESCRIPCION DE REGISTRO | | | | |
| Pos. | NOMBRE | TIPO | LARGO | DESCRIPCION |
| 1 | COD-PETICION-DPTO | | | |
| | FAC-CODIGO | ? | 1 | CODIGO FACULTAD |
| | CAR-CORRELATIVO | ? | 2 | CORRELATIVO CARRARRERA |
| | CUR-ARO | ? | 2 | ARO DE LA CURRICULA |
| | DEP-CODIGO | X | 2 | CODIGO DEPARTAMENTO |
| | ASI-CORRELATIVO | ? | 3 | CORRELATIVO ASIGNATURA |
| 11 | PRE-ALU-EST | ? | 3 | ALUMNOS ESTIMADOS |
| 14 | PRE-ALU-REA | ? | 3 | ALUMNOS REALES |
| 17 | PRE-BRUPOS-CAT-EST | ? | 2 | GRUPOS CATEDRAS ESTIMADOS |
| 19 | PRE-BRUPOS-PRA-EST | ? | 2 | GRUPOS PRACTICA ESTIMADOS |
| 21 | PRE-BRUPOS-LAB-EST | ? | 2 | GRUPOS LABORATORIOS ESTIMADOS |
| 23 | PRE-BRUPOS-CAT-REA | 99V99 | 4 | GRUPOS CATEDRAS REALES |
| 27 | PRE-BRUP-PRA-REA | 99V99 | 4 | GRUPOS PRACTICA REALES |
| 31 | PRE-BRUP-LAB-REA | 99V99 | 4 | GRUPOS LABORATORIOS REALES |
| 39 | PRE-MARCA-PROCESADA | X | 1 | MARCA DE PETICION PROCESADA |
| 36 | PRE-FECHA-INGRESO | DATE | 6 | FECHA INGRESO DE LA PETICION |
| 42 | SEC-CLAVE | X | 1 | CLAVE DE SECCION |

ARBOL DE MENUES

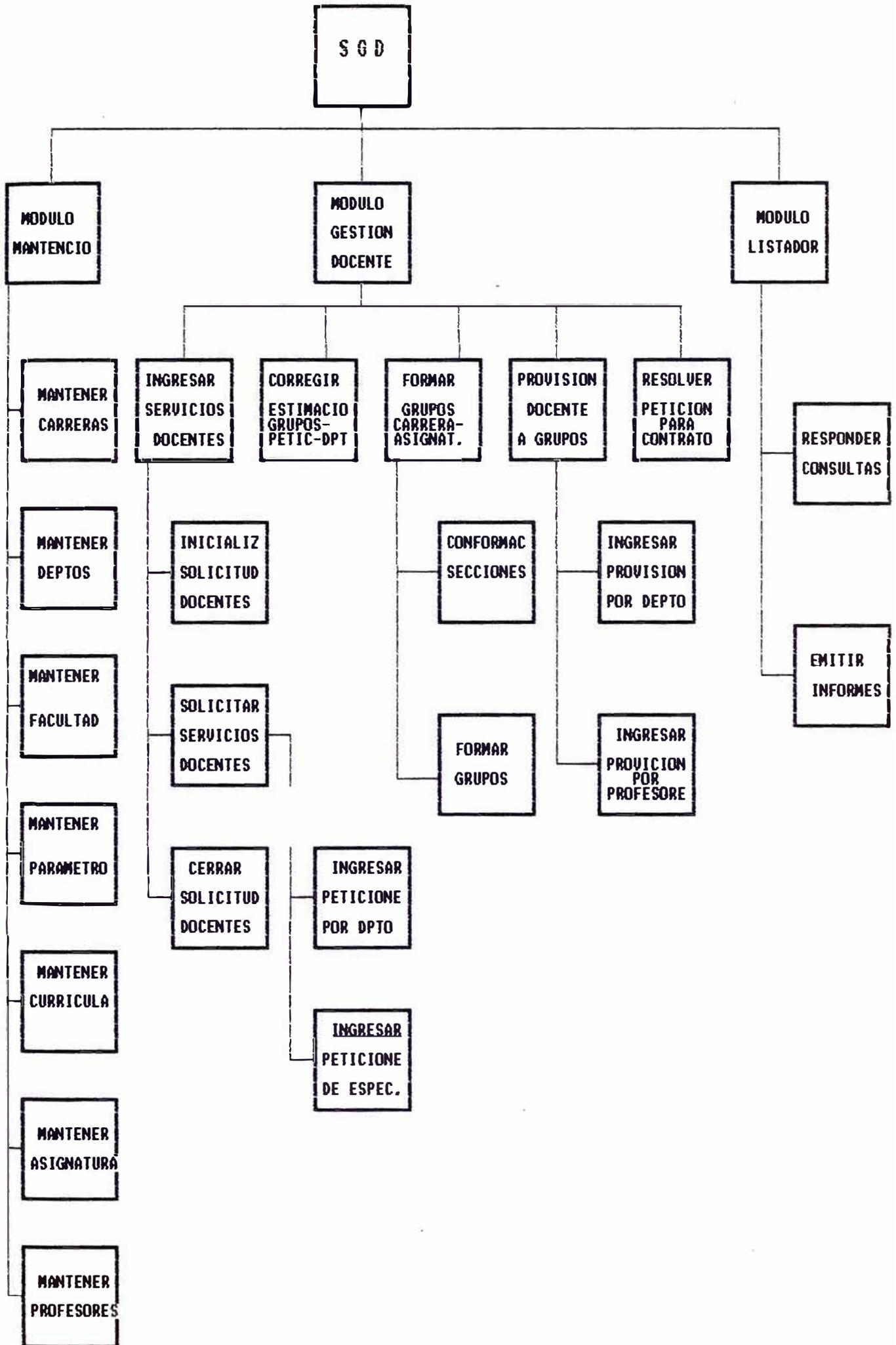


TABLA DE PROCESOS

| PROCESOS | ARCHIVOS | | | | | | | | | | | | | I N F O R M E S | C O N S U L T A | |
|---------------------|--|---|--|----------------------------|--------------------------------------|--|--|--|----------------------------|----------------------------|--------------------------------------|--|---|--------------------------------------|--------------------------------------|---|
| | A S I G N A T U R I A S | I N S C R I P T I O N E S | F A C U L T A D E S | D E P T O S | C A R R E R A S | C U R R I C U L O S | P R O F E S O R E S | P A R A M E T R O S | C U R S O S | G R U P O S | D O C E N C I A | P E T I C I O N - D E P T O | P E T I C I O N - E S P E C I A L | | | S E C C I O N E S |
| MANTENER CARRERA | | | | | X | | | | | | | | | | | |
| MANTENER DEPARTAMEN | | | | X | | | | | | | | | | | | |
| MANTENER FACULTADES | | | X | | | | | | | | | | | | | |
| MANTENER PARAMETROS | | | | | | | | X | | | | | | | | |
| MANTENER CIRRICULA | | | | | X | X | | | | | | | | | | |
| INGRESAR PET.DPTO. | X | | | | X | X | | X | | | X | | X | | | |
| INGRESAR PET.ESPECI | X | | | | X | X | | X | | | | X | | | | |
| RESOLVER PET.CONTRA | | | | | | | | | | | X | X | X | | | |
| FORMAR GRUPOS CARRE | | | | | | | | | | | X | | | | X | |
| FORMAR GRUPOS ASIGN | | | | | | | | | X | X | | | | | X | |
| INGRESAR PROVIS.DOC | X | | | X | | X | X | | X | X | X | | | | | |
| CORREGIR ESTIMACION | | X | | | | | | X | | | | | | | | |
| RESPONDER CONSULTAS | | | X | X | X | X | X | X | X | X | X | X | X | | | X |
| EMITIR INFORMES | | | X | X | X | X | X | X | X | X | X | X | X | | X | |

CONCLUSIONES Y RECOMENDACIONES

Los objetivos planteados inicialmente para este proyecto, se cumplieron. La metodología de Ingeniería Reversa aplicada al Sistema de Información Gestión Docente, permitió reusar, en un regular porcentaje, el diseño del sistema anterior, lo que conllevó a un ahorro en el costo de tiempo en la etapa de análisis y diseño del sistema y permitió concentrarse en otras etapas que requerían una mayor inversión de tiempo.

Una gran ventaja obtenida con el desarrollo de esta aplicación utilizando la metodología de Ingeniería Reversa, fue si bien no obviar, reducir en forma considerable el tiempo invertido en las actividades de entrevistas con el usuario de sistemas.

El desarrollo de esta aplicación de migración permitió mostrar e ilustrar la funcionalidad de esta metodología, y los diferentes procesos participantes que se utilizaron para lograr el completo entendimiento de la funcionalidad y semántica del sistema en cuestión y dominio del problema. También se pudo realizar una real actualización de la documentación que se manejaba. Otro logro es la productividad alcanzada por concepto de reusabilidad.

Como se pudo apreciar del trabajo realizado, las mantenciones o modificaciones, que sufrió este sistema, no fueron reflejadas en algunos casos y en otros se registraron arcialmente, en la documentación de diseño; los cuales fueron

superados por esta metodología, que es una buena técnica para lograr documentación actualizada.

Existe una gran carencia de documentación con respecto al tema de Rejuvenecimiento del software, específicamente sobre Ingeniería Reversa. Se podría pensar que ocurre esto, porque son; temas nuevos que se están recién incorporando en los productos de software, como son: el caso de las herramientas CASE.

Una de las dificultades al aplicar esta metodología, es la de no contar con herramientas de tecnología CASE, lo cual nos hubiese ahorrado un tiempo sustancial en la obtención de las especificaciones de diseño. Por lo cual todo el proceso de ingeniería inversa se desarrolló en forma manual.

A pesar de contar con cierta documentación de diseño se encontró la gran dificultad de entender la estructura general de los programas desarrollados en Cobol ya que estos tenían una lógica no estructurada y en algunos casos se encontraron modificaciones que no se reflejaban en la documentación.

Se tuvo que estudiar con mayor detenimiento la documentación de los programas fuente y del diseño de la estructura de datos, que pertenecen a un nivel de abstracción relativamente bajo, para lograr un alto nivel de abstracción como es el diagrama Entidad-Relación, que nos permite representar gráficamente los objetos de datos como Asignatura, Facultad, Profesores, Currícula, etc. y las relaciones entre estos.

Ya que la aplicación de Ingeniería reversa demanda un enorme esfuerzo, se recomienda en su aplicación, seleccionar de la biblioteca de programas, aquellos que más se estén usando, y que probablemente se seguirán usando, estimar el coste anual de mantenimiento de estos incluyendo: la corrección de errores, la adaptación al entorno y las mejoras

funcionales para que finalmente se les aplique la metodología. Estos pasos no fueron seguidos estrictamente en el presente trabajo, ya que, se migrará a una nueva plataforma de trabajo, MicroVAX 3100 y Base de Datos Ingres, por lo tanto se tiene que rehacer todos los programas nuevamente.

Para futuros desarrollos se recomienda, ver los módulos del sistema como elementos que inevitablemente, estarán sujetos a cambios, con lo cual, serán sustancialmente mayores las posibilidades de producir un sistema fácilmente mantenible.

Si bien la metodología de Ingeniería Reversa no tiene el tradicional sentido de desarrollo al cual se está acostumbrado a trabajar, sin embargo, es una buena forma que se está incluyendo en la mayoría de las herramientas Case actuales, como por ejemplo, "Bachman" de Bachman Information System Incorporation, "Structured Retrofit" de Peat Marwick Advanced technology, "Via/Insight" de Viasoft Incorporation, etc.

ANEXO No 1 : INGENIERÍA**REVERSA Y LAS LEYES DE PROPIEDAD****INTELECTUAL****1.1 INTRODUCCION**

Los valiosos productos de software disponibles en el mercado hoy en día tienen Derechos de Copia. Las leyes de Derecho de Copia han llegado a ser el principal terreno de batalla del debate sobre si la tecnología de Reingeniería Reversa puede ser usada sobre sistemas de otras empresas. La política sobre la cual este debate legal se sustenta es si éste infringe los Derechos de Copia de las otras empresas, al hacer una copia desautorizada de un sistema o programa con Derechos de Copia. Algunos hombres de leyes creen que esto hace infringir los Derechos de Copia y otros no.

DEFINICION DE TERMINOS

Definiciones : Existe un acuerdo general sobre un sistema con Derecho de Copia, el cual es realizar una prueba de Caja Negra de sus operaciones bajo condiciones e inferencias lógicas para analizar sus salidas. La cual es conducta legal y no infringe los Derechos de Copia.

1.2 INTERPRETANDO LAS LEYES

Ingeniería Reversa es generalmente considerada una forma legal para tomar los Secretos de Negocios, en otras disciplinas de Ingeniería, el pragmatismo también piensa que

esta podría ser considerado como una apropiada conducta competitiva en el software.

Los Derechos de Copia en la Ingeniería Reversa de software, si son ilegales por: tomar el código assembler e interpretarlo a un más alto nivel de abstracción, para hacer algunas modificaciones y usarlo comercialmente, de igual forma se puede considerar para el código compilado y otros procesos similares que son siempre una forma ilegal.

1.3 EL USO LEGAL

Se lista un conjunto de factores para ser considerados en determinar si un uso es legal o ilegal, incluyendo:

- El propósito de seguridad en usar el trabajo protegido
- La naturaleza del trabajo con los Derechos de Copia
- La cantidad y volumen de los que es tomado
- El potencial para dañar al mercado del trabajo protegido.

1.4 LOS SECRETOS DE NEGOCIOS

Entre las partes más importantes del producto software están los algoritmos, conceptos y técnicas usadas en el software. Las partes tradicionales no cubiertas por los Derechos de Copia, deben ser protegidos como Secretos de Negocios.

1.5 ¿ QUE SE PUEDE REALIZAR ?

Hay muchas cosas que se pueden realizar con el conocimiento obtenido después de aplicada Ingeniería Reversa.

1.5.1 HACER MODIFICACIONES

Existen los Derechos de Copia dando al propietario el derecho a controlar todas las modificaciones. Existen normas especiales que deja a quienes compran el software realizar algunas modificaciones a éste.

Se permite modificaciones que hacen al software más útil para el propósito que fué adquirido.

1.5.2 DESARROLLAR UN SOFTWARE SIMILAR

A diferencia de modificar el software, otras cosas que se podría hacer con los resultados de Ingeniería Reversa, por ejemplo, podrían ser desarrolladas otras piezas de software que son parecidas, pero no idénticas a una que fue aplicada la metodología de Ingeniería Reversa. Sin embargo, Ingeniería Reversa desarrolla un producto de software competitivo, el cual puede afectar el mercado del software original.

Por ejemplo, Ingeniería Reversa toma solamente ideas, algoritmos, técnicas, lógicas y elementos funcionales del software original e incorpora éstos en una nueva pieza de software, no infringiendo las leyes de propiedad intelectual.

1.6 DIFERENTES LEYES E INGENIERIA REVERSA

A continuación se detallan algunas leyes que generan un debate sobre la legalidad de Ingeniería Reversa, estas son:

Los propietarios de los Secretos de Negocios tienen derechos a ser protegidos contra la malversación de estos mismos. Las dos más comunmente reconocidas formas para malversar tales secretos son: por quiebre de una relación confidencial que se origina cuando los propietarios dan el secreto en confidencia a otro y por uso de significado

impropio (semejante al espionaje industrial) para obtener el Secreto de Negocio.

Ingeniería Reversa no se ha alzado generalmente como un área de problema en la ley de patente, por la simple razón que la ley de patente siempre ha requerido que un inventor revele una considerable cantidad de detalle sobre la invención para obtener una patente. Las aplicaciones de patentes deben contener "una descripción escrita de la invención, y del modo y proceso de hacer y usar éste, en forma completa, clara, y precisa.

**ANEXO No. 2 : ASPECTOS
GENERALES DE LA HERRAMIENTA**

INGRES

2.1 INTRODUCCION

Ingres es un Sistema Gestor de Bases de Datos Relacionales, que puede ser usado sobre un amplio rango de equipos: mainframe, minicomputers y microcomputers para definir, manipular y proteger los datos; y para construir aplicaciones por usuarios.

Ingres ha sido creado para generar soluciones a la empresa, para los actuales desafíos de la administración de la información y para proveer un soporte amplio.

Ingres comenzó sus operaciones en el año 1980, debido a la creciente necesidad de mejorar la administración de los recursos informáticos en las empresas, utilizando un concepto de Base de Datos que se fundamenta en el Modelo Relacional. Desde el principio Ingres fue diseñado para optimizar el retorno de la inversión en la informática que realizan las empresas que incluye equipamiento, datos y personal. Desde entonces Ingres se ha posicionado dentro del sector de las Bases de Datos como un líder en tecnología, posición que se ha mantenido por una constante y continua innovación.

2.2 ¿ QUE ES INGRES ?

Ingres es un Relational Database Management System.

Es una Database porque es un conjunto de información relacionada que Ingres almacena dentro del equipo donde se encuentra trabajando. La unidad básica de información que la Base de Datos almacena son llamados datos, los cuales se encuentran agrupados en tablas.

El Management significa que Ingres adopta el cuidado de almacenar los datos de un usuario Ingres, dejando al usuario ingresar datos dentro de su Base de Datos como tambien recuperar datos de ésta.

Ingres es un System porque se pueden combinar sus diferentes componentes para organizar y accesar sus datos de tal forma que el trabajo se facilite para el usuario. Ingres presenta una variedad de herramientas para funciones como: crear reports, diseñar Forms para ingreso de datos y para facilitar la recuperación de información desde la Base de Datos.

Finalmente Ingres es Relational, porque él almacena los datos en las tablas facilitando hacer revisiones a diferentes partes de la información, que a veces corresponde a información de una o más tablas - uniones entre dos o más tablas-. Una tabla contiene columnas (campos) para cada tipo de datos y filas (registros) de toda la información que se ingresa en la tabla.

2.3 DESCRIPCION TECNICA DEL SOFTWARE

2.3.1 ESTANDARES

La arquitectura Ingres está basada en una estricta adherencia a los estándares de la industria, tales como OSI y

ANSI-SQL, debido a que se reconoce la importancia de la existencia y aceptación de estos estándares. Ingres está basado en SQL estándar de ANSI, por razones históricas, esta herramienta también ofrece QUEL, un lenguaje de consulta interactivo.

2.3.2 PORTABILIDAD DE LAS APLICACIONES

La portabilidad de las Bases de Datos y sus aplicaciones es uno de los fundamentos de Ingres, pilar conceptual que se extiende a la conectividad multiambiente de Ingres/Net y las Bases de datos distribuidas en Ingres/Star.

Ingres ofrece utilitarios amigables para el traspaso de la Base de Datos, dentro de la misma máquina o entre máquinas con distinta arquitectura o sistema operativo. Por ejemplo, el traspaso de códigos fuentes se efectúa usando un utilitario denominado COPYAPP. Las diferentes arquitecturas que mantienen Ingres son: computadores personales, equipo Unix, equipo Vax/VMS y otros.

2.3.3 UTILITARIOS DE ADMINISTRACION DE INFORMACION

Ingres provee una gran cantidad de utilitarios de administración y carga de datos, podemos mencionar algunos:

| UTILITARIOS | F U N C I O N |
|-------------|--|
| ABF | Invoca a Aplicación by Forms |
| CATALOGDB | Lista las BD que pertenece al usuario |
| COPY TABLE | Permite copiar los datos de una tabla |
| COPYAPP | Copia una Aplicación de una BD a otra |
| COPYFORM | Copia un Forms desde una BD a otra |
| INFODB | Provee información de la Base de Datos |
| INGMENU | Invoca al ambiente Ingres/Menu |

2.3.4 INTERFAZ A LENGUAJES DE PROGRAMACION

Ingres ofrece una gran variedad de pre-compiladores, los cuales permiten la creación y compilación de aplicaciones creadas en lenguajes de tercera generación. Esta herramienta incluye pre-compiladores para los lenguajes C, Cobol, Fortran, Ada, etc.

2.3.5 HERRAMIENTAS PARA EL DESARROLLO DE APLICACIONES

Ingres posee herramientas de ayuda a la toma de decisiones que ayudan a los usuarios a obtener rápidamente la información que requieren, liberando a los programadores de tareas que pueden realizar los usuarios directamente, y así concentrarse en labores asistidas por menús que permiten realizar consultas, cargar y actualizar información, generar reportes, crear pantallas, crear interfaces, etc..

BIBLIOGRAFIA

- Título : IEEE Software, "ReverseEngineering and Design Recovery: A taxonomy", January 1990
Autor : Eliot J. Chikofsky - James H. Cross
- Título : IEEE Software, "Reverse Engineering Someone Else's Software: Is it Legal ?", January 1990
Autor : Pamela Samuelson
- Titulo : IEEE Computer Society, "Design Recovery" July 1989
Autor : Ted J. Biggerstaff
- Título : IBM Systems Journal, "Re-Engineering Software: A case study", 1990
Autor : Robert N. Britcher
- Título : ACM Software Engineering Notes, "A view on three R's: Reuse, Re-Engineering, and Reverse-Engineering", July 1991
Autor : Don Yu
- Título : ACM Software Engineering Notes, "Reveng: A Cost-Effective approach to Reverse-Engineering", October 1992
Autor : Xavier A. Debest-Rudiger Knoop-Jurgen Wagner
- Título : "Technology Assessment on Software Re-Engineering, Contel Technology Center Technical Report", February 1990
Autor : S. A. Bohner

Título : Informática, "Ingeniería Reversa al rescate... y al ataque", Abril 1990

Autor : Pedro Hepp

Título : Paper, "Reingeniería de Software: Transformación de código fuente"

Autor : David R. Stevenson

Título : Journal System Software, "Reverse Engineering process, Design Recovery and Structure Charts", 1992

Autor : P. Benedusi - A. Cimitile - U. De Carlini

Título : "Using Ingres Through Forms and Menus"

Autor : Ingres

Título : "Ingres/Vision User's Guide"

Autor : Ingres

Título : "Ingres Report-Writer Reference Manual"

Autor : Ingres

Título : "Ingres 4GL Reference Manual"

Autor : Ingres

Título : "ingress/SQL Reference Manual"

Autor : Ingres

Título : "Ingres Database Administrator's Guide"

Autor : Ingres