

11  
2ij



**UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO**

ESCUELA NACIONAL DE ESTUDIOS PROFESIONALES  
"ACATLAN"

**DISEÑO DE UN SISTEMA PARA EL  
CONTROL DEL SERVICIO SOCIAL**



**T E S I S**

QUE PARA OBTENER EL TITULO DE:

LICENCIADO EN MATEMATICAS  
APLICADAS Y COMPUTACION

P R E S E N T A :

**MARIA EUGENIA TELLEZ SANTOS**

ASESOR DE TESIS :

M. EN C. SERGIO VICTOR CHAPA VERGARA

**TESIS CON  
FALLA DE ORIGEN**

MEXICO, D. F.

1991



## **UNAM – Dirección General de Bibliotecas Tesis Digitales Restricciones de uso**

### **DERECHOS RESERVADOS © PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL**

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

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

## INDICE

	Página
OBJETIVO	1
INTRODUCCION	2
CAPITULO 1. BASES DE DATOS	4
1.1 Organización de los Datos	4
1.2 Estructura Jerárquica	8
1.3 Estructura en Red	11
1.4 Estructura Relacional	14
1.5 Algebra Relacional y Cálculo Relacional	20
1.6 Normalización	24
CAPITULO 2. ANALISIS DE LOS REQUERIMIENTOS DE INFORMACION DEL DEPARTAMENTO DE SERVICIO SOCIAL	30
2.1 Técnica de Anlisis Estructurado de Sistemas	30
2.2 Análisis del Flujo de Datos del Departamento de Servicio Social	31
2.3 Desarrollo de los Diagramas de Flujo de Datos	41
CAPITULO 3. DISEÑO DEL SISTEMA	50
3.1 Diseño Descendente (Top-down) de la Estructura del Sistema	50
3.2 Diseño de la Base de Datos	57
CAPITULO 4. DESARROLLO DEL SISTEMA	64
4.1 Introducción al Lenguaje de Flujograma	64
4.2 Desarrollo de los Programas del Sistema	65

CONCLUSIONES

84

BIBLIOGRAFIA

85

APENDICE

86

## OBJETIVO

El objetivo principal de la presente tesis consiste en realizar el análisis, diseño y desarrollo de sistemas (en este caso, un sistema que apoye el registro y control de los prestadores y los programas de servicio social) utilizando algunas técnicas de Ingeniería de Software, aplicar la teoría de Bases de Datos y presentar una nueva metodología para el desarrollo de sistemas.

Para llevar a cabo dicho desarrollo se utilizara el Lenguaje de Flujoograma de S. Chapa ya que, debido a su naturaleza icónica basada en el concepto de flujo de datos, facilita la representación y comprensión de procedimientos, así como la generación automática de líneas de código en algún lenguaje tradicional.

## INTRODUCCION

Una de las entidades de mayor importancia y movimiento administrativo dentro de la Escuela Nacional de Estudios Profesionales Acatlán, es el Departamento de Servicio Social. Este departamento es el responsable de la captación, evaluación y difusión de los programas de servicio social, así como del registro, control y evaluación de los prestadores del mismo. En adición a esto, el Departamento de Servicio Social cuenta con una sección de Bolsa de Trabajo, la cual presta sus servicios a la comunidad universitaria.

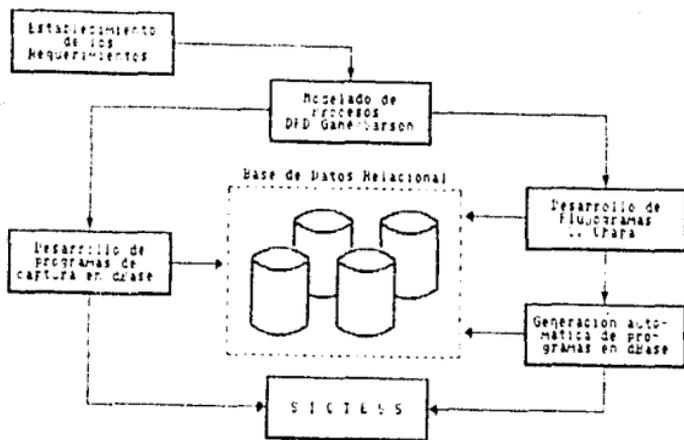
Es aquí donde surge la necesidad de diseñar un sistema automatizado que cubra satisfactoriamente los requerimientos de información del Departamento en su parte administrativa, manejando en forma eficiente el registro y control de los prestadores y de los programas del servicio social.

Dicho sistema utilizará una metodología de diseño e implementación del modelo de flujo de datos desde un punto de vista único. Para el primero, se considera el modelo de procesos basado en el flujo de datos de Gane/Sarson de requerimientos en Ingeniería de Software. El segundo utiliza el lenguaje de flujograma de S. Chapa para el desarrollo de los principales programas, los cuales podrán ser transformados posteriormente en código en dBase III. Los programas de captura deberán estar directamente desarrollados en dBase III Plus, debido a los múltiples controles de validación que contienen, y que son difíciles de implementar en el lenguaje de flujograma.

El sistema diseñado para cubrir tales necesidades denominado SICTESS (Sistema de Control de Trámites y Evaluación del Servicio Social) está fundamentado en técnicas de análisis estructurado de sistemas (también conocidas como Ingeniería de Software) incorporando el desarrollo de Diagramas Lógicos de Flujo de Datos para el análisis del sistema (capítulo 2), un diseño descendente (top-down) y modular (capítulo 3), y el desarrollo del mismo en un lenguaje relacional (capítulo 4).

El capítulo 1 presenta una descripción de los diferentes modelos de bases de datos que se establecen para el diseño de un Sistema Manejador de Bases de Datos (DBMS). Se dan los elementos principales que deben constituir un modelo relacional.

En el capítulo 2 se analizan en forma detallada las necesidades y los problemas por resolver que presenta el Departamento de Servicio Social y el flujo de información a través de los estados que lo componen utilizando diagramas de flujo de datos (DFD).



Esquema general de la tesis

El capítulo 3 comprende el diseño estructurado de los módulos que componen el sistema, así como el diseño de la base de datos, siguiendo un esquema relacional

En el capítulo 4 se elaboran los principales programas que componen el sistema utilizando el lenguaje de flujoograma desarrollado por S.Chapa, el cual consta de un lenguaje gráfico con el modelo de flujo de datos que unifica lo planteado en los requerimientos con el modelo de flujo de datos de Gane/Sarson.

Por último se presentan las conclusiones obtenidas del presente trabajo, la bibliografía y un apéndice el cual contiene los programas fuente de captura desarrollados en dBase III Plus

Es importante resaltar la principal contribución de la presente tesis, la cual consiste en el desarrollo del sistema en un nivel alto de abstracción, incorporando una nueva metodología para el desarrollo de sistemas.

## CAPITULO 1: BASES DE DATOS

Aún cuando no existe una definición formal de lo que es una Base de Datos, puede decirse que una base de datos eficiente es una colección de datos interrelacionados almacenados en conjunto sin redundancias perjudiciales o innecesarias, cuya finalidad es la de servir a una o varias aplicaciones a la vez, de la mejor manera posible. Los datos se almacenan de modo que resulten independientes de los programas que los usan; se emplean métodos bien determinados para incluir datos nuevos y modificar o extraer los datos ya existentes. Se dice que un sistema comprende una colección de bases de datos cuando estas son totalmente independientes desde un punto de vista estructural.

Las características básicas con que debe contar una base de datos son:

- no redundancia de datos
- independencia de datos
- interconectividad
- seguridad
- accesibilidad en tiempo real

### 1.1 ORGANIZACION DE LOS DATOS

Existen 3 tipos de organización de los datos en el manejo de base de datos, los cuales son:

#### 1. Organización de Archivos.

Esta parte se preocupa de la vista de los datos, tal como lo percibe el programador de aplicaciones.

#### 2. Organización Lógica Global de los Datos.

Se preocupa de la organización general de la base de datos, de la cual pueden derivarse múltiples organizaciones de archivos.

#### 3. Organización Física de Almacenamiento.

Tiene como objetivo la representación y la distribución física de los datos y la organización de estos en las unidades de almacenamiento (índices, punteros, cadenas, etc.).

El presente trabajo está dedicado a la organización lógica de los datos dentro de la base de datos.

### Objetivos Primarios de la Organización de la Base de Datos

A continuación se presenta una relación de los objetivos básicos que se persiguen al tener organizados los datos dentro de la base de datos.

1. Los datos podrán utilizarse de múltiples maneras.  
Diferentes usuarios, que perciban inclusive en forma diferente los datos, pueden emplearlos de distintas maneras.
2. Se protegerá la inversión intelectual.  
No será necesario rehacer los programas y las estructuras lógicas existentes (que pueden representar varios años/hombre de trabajo) cuando se modifique la base de datos.
3. Bajo costo.  
Bajo costo de almacenamiento y uso de los datos, y minimización del costo de los cambios.
4. Menor proliferación de datos.  
Las necesidades de las nuevas aplicaciones se satisfarán con los datos existentes más que creando nuevos archivos, evitándose así la excesiva proliferación de datos que se advierte en la mayoría de las bibliotecas de cintas hasta hace pocos años.
5. Desempeño.  
Los pedidos de datos se atenderán con la rapidez adecuada según el uso que de ellos habrá de hacerse.
6. Claridad.  
Los usuarios sabrán qué datos se encuentran a su disposición y los comprenderán sin dificultad.
7. Facilidad de uso.  
Los usuarios tendrán fácil acceso a los datos. Las complejidades internas son ajenas al usuario, gracias al sistema de administración de la base.

8. **Flexibilidad.**  
Los datos podrán utilizarse o analizarse de manera flexible, con diferentes caminos de acceso.
9. **Rápida atención de interrogantes no previstos.**  
Las solicitudes espontáneas de información se atenderán sin necesidad de escribir un programa de aplicación (lo cual significa un cuello de botella debido a la pérdida de tiempo), sino utilizando un lenguaje de alto nivel para consultas o generación de reportes.
10. **Facilidad para el cambio.**  
La base de datos puede crecer y variar sin interferir con las maneras establecidas de utilizar los datos.
11. **Precisión y coherencia.**  
Se utilizarán controles para garantizar la precisión. El sistema evitará las versiones múltiples de los mismos elementos de datos con diferentes estados de actualización.
12. **Reserva.**  
Se evitará el acceso no autorizado a los datos. Los datos podrán estar sujetos a diferentes restricciones de acceso para diferentes usuarios.
13. **Protección contra pérdida o daño.**  
Los datos estarán protegidos contra fallas y catástrofes, y contra delincuentes, vándalos, incompetentes y personas que intenten falsearlos.
14. **Disponibilidad.**  
Los datos se hallarán inmediatamente disponibles para los usuarios casi todas las veces que los necesiten.

#### **Objetivos Secundarios** (para facilitar el logro de los objetivos primarios)

1. **Independencia física de los datos.**  
El hardware de almacenamiento y las técnicas físicas de almacenamiento podrán modificarse sin necesidad de cambiar los programas de aplicación.

2. **Independencia lógica de los datos.**  
Podrán agregarse nuevos ítems de datos, o expandirse la estructura lógica general, sin que sea necesario reescribir los programas de aplicación existentes.
3. **Redundancia controlada.**  
Los elementos de datos serán almacenados una sola vez, excepto cuando existan razones técnicas, económicas o de seguridad que aconsejen el almacenamiento redundante.
4. **Adecuada rapidez de acceso.**  
Los mecanismos de acceso y los métodos de direccionamiento serán lo suficientemente rápidos en función de los usos previstos.
5. **Adecuada rapidez de exploración.**  
La conveniencia y necesidad de la exploración espontánea se incrementarán en la medida que se difunda el uso interactivo de los sistemas.
6. **Normalización de los datos dentro de un organismo.**  
Se necesita un acuerdo interdepartamental sobre los formatos y las definiciones de datos. La normalización entre departamentos es indispensable porque de otro modo ellos crearían datos incompatibles.
7. **Diccionario de datos.**  
Se necesita un diccionario de datos que defina todos los elementos de datos.
8. **Interface de alto nivel con los programadores.**  
Los programadores de aplicaciones deben disponer de medios sencillos para pedir datos y estar aislados de las complejidades internas de organización y direccionamiento de los archivos.
9. **Lenguaje del usuario final.**  
Un lenguaje de consulta de alto nivel o un lenguaje para la generación de reportes permitirán que los usuarios finales se vean libres de tener que escribir un programa de aplicación convencional.

**10. Controles de integridad.**

Siempre que sea posible, se recurrirá a revisar los límites y otros controles para asegurar la exactitud de los datos.

**11. Fácil recuperación en caso de falla.**

Recuperación automática sin pérdida de datos.

**12. Afinación.**

La base de datos debe ser afinable para mejorar su desempeño sin exigir la reescritura de los programas de aplicación.

**13. Ayudas para el diseño y la supervisión.**

Ayudas que permitan al diseñador o al administrador de datos predecir y optimizar el desempeño.

**14. Migración o reorganización automática.**

Migración de datos u otra reorganización física previstas para mejorar el desempeño.

A continuación se presentan los tipos de estructuras a que se ajustan las relaciones en una base de datos. Estas estructuras pueden ser del tipo jerárquico, en red o relacionales. Cabe mencionar que tanto las estructuras jerárquicas como las de redes se dejan descomponer en grupos de archivos planos (matriz bidimensional donde cada registro tiene el mismo número de campos) con elementos de datos redundantes en ocasiones.

## 1.2 ESTRUCTURA JERARQUICA

Esta estructura se representa por medio de árboles.

Todo árbol está compuesto por una jerarquía de elementos denominados **nodos**. El nivel más alto de la jerarquía tiene un solo nodo denominando **raíz**.

Con excepción de la raíz, todo nodo está vinculado a otro nodo de nivel superior llamado **padre**. Ningún nodo puede tener más de un padre. Por otro lado, todo elemento puede tener uno o más elementos relacionados, en un nivel más bajo denominados **hijos**. Los elementos que se encuentran en las puntas de las ramas (es decir, aquellos nodos que no tienen hijos) se llaman **hojas**.

En la Fig. 1.1, el nodo 1 es la raíz. Los elementos 5 y 6, 8-12 y 14-22 son hojas.

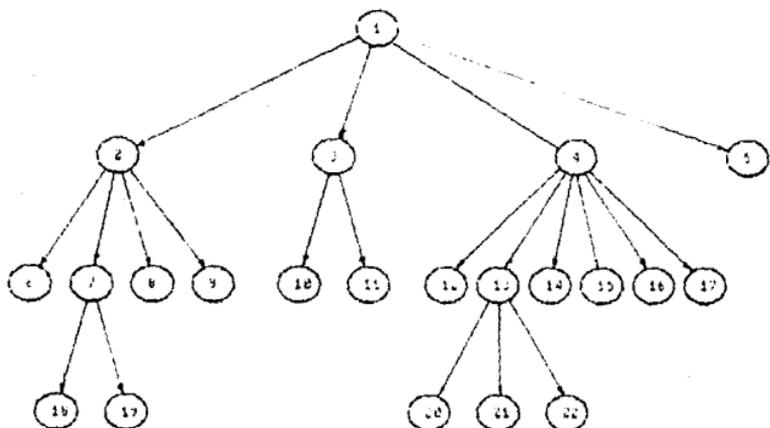


FIG. 1.1 Estructura jerárquica

Todo árbol puede ser descrito como una jerarquía de nodos con relaciones binodales, de tal modo que:

1. El más alto nivel de la jerarquía tiene un solo nodo llamado raíz.
2. Los nodos restantes se reparten en  $m = 0$  conjuntos disjuntos (es decir, no conectados)  $T_1, \dots, T_m$  y cada uno de estos conjuntos constituye a su vez un árbol. Los árboles  $T_1, \dots, T_m$  se llaman subárboles de la raíz.

Dentro de la terminología utilizada para describir árboles tenemos:

<u>Nombre</u>	<u>Descripción</u>	No. de incidencias en el Arbol de la Fig. 1.1
Altura:	No. niveles	4
Momento:	No. nodos	22
Peso:	No. hojas	16
Raíz:	No. raíces	1
Grado del nodo:	No. hijos	Nodo 2 grado 4
Dimensión:	No. hijos que tienen un nodo padre	Fam. del nodo 4 de dim. 6
Cuenta:	No. nodos que tiene el nivel	Nivel 4 cuenta 5

El nombre de Base de Datos Jerárquica se aplica a los archivos que presentan relaciones de tipo árbol entre sus registros.

En la Fig. 1.2 se muestra una base de datos de tipo jerárquico con 2 tipos de registro.

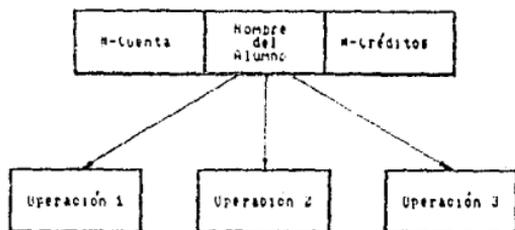


FIG. 1.2 Base de datos jerárquica con dos registros

En la Fig. 1.3 se muestra una base de datos jerárquica de 4 niveles.

La aplicación de este tipo de estructura es útil en un caso sencillo como este, pero no así para relaciones más complejas que no sean de tipo ramificado, ya que algunos tipos de registro pueden tener más de un padre. Además, este tipo de estructura dificulta la actualización de datos debido a la serie de recorridos que tienen que realizarse para alcanzar un nodo específico, por lo que tiende a caer en la redundancia de información.

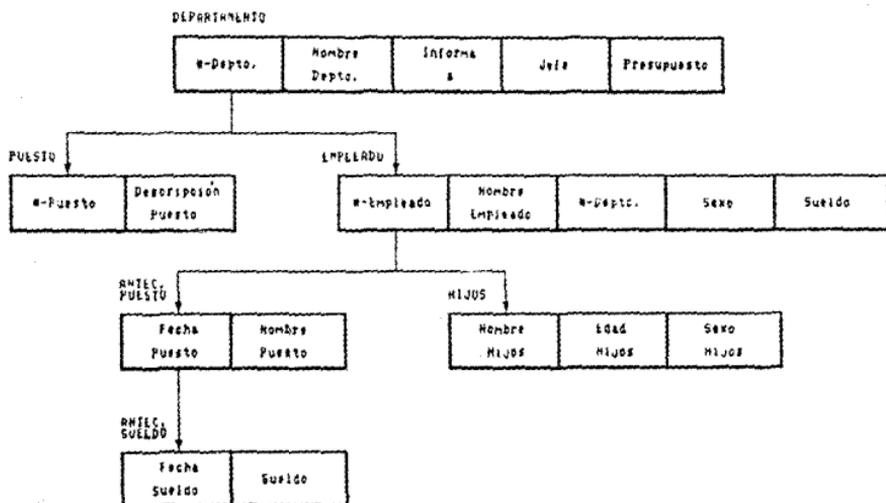


FIG. 1.3 Base de datos jerárquica de 4 niveles

### 1.3 ESTRUCTURA EN RED

Este tipo de estructura se utiliza en caso de que un nodo hijo tenga más de un padre, ya que cualquier componente puede relacionarse con cualquier otro.

Es importante hacer notar que toda estructura en red puede reducirse a una forma más simple (de árbol) introduciendo cierta redundancia. Esto se muestra en la Fig. 1.5, la cual se obtiene a partir de la Fig. 1.4a.

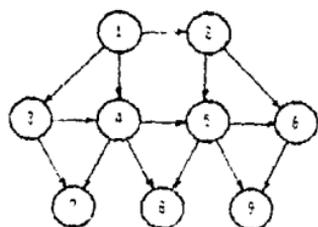
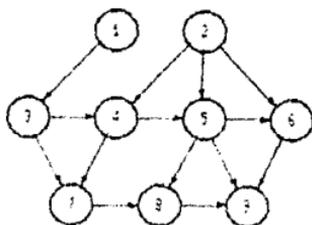


FIG. 1.4 a) Red con un nodo origen



b) Red con dos nodos origen

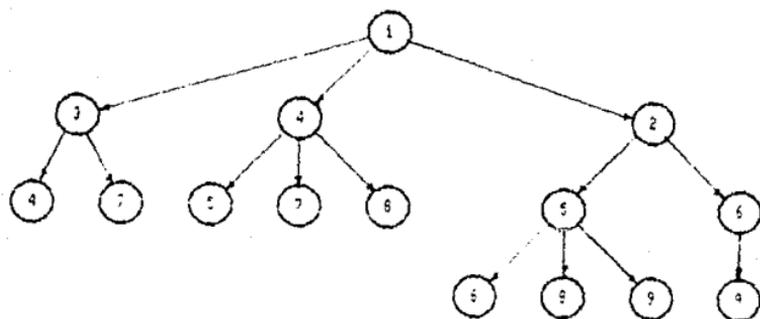


FIG. 1.5 Árbol proveniente de la red de la fig. 1.4a

En algunos casos, esta redundancia no es mucha y se tolera sin problemas, pero en otros es excesiva pues lleva a tener que elaborar varias estructuras de árbol, como lo muestra la Fig. 1.6, la cual proviene de la Fig. 1.4b.

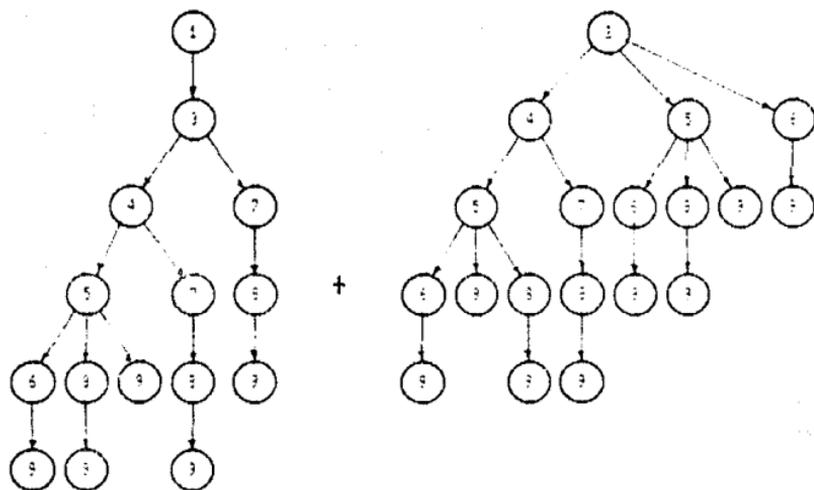


FIG. 1.6 Árboles provenientes de la red de la fig. 1.4b

La Fig. 1.7 muestra una estructura tipo red con 2 registros.

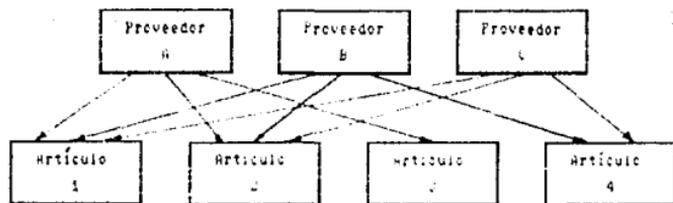


FIG. 1.7 Estructura de red con 2 registros

#### 1.4. ESTRUCTURA RELACIONAL

Aún cuando las estructuras jerárquicas y de red presentan algunas facilidades, su complejidad aumenta a medida que se incrementa el número de elementos y sus relaciones entre ellos, creando así una "maraña" de relaciones rebuscadas e inflexibles.

Debido a lo anterior, se hace necesaria la aplicación de una manera de escribir los datos que:

1. pueda ser entendida fácilmente por los usuarios que no tienen preparación previa como programadores.
2. que haga posible ampliar la base de datos sin modificación de la estructura lógica existente, y por lo tanto, sin modificación de los programas de aplicación;
3. que permita la máxima flexibilidad en la formulación de interrogantes de forma no prevista o espontánea.

Una de las maneras más sencillas de representar datos al usuario, es la que se basa en tablas bidimensionales ( renglones y columnas). De la misma forma en que toda estructura en red se reduce a una representación en árbol manteniendo cierta redundancia, cualquier representación se puede llevar a la forma tabular o de archivo plano bidimensional, admitiendo una adecuada parte de redundancia.

La normalización (proceso generado por E. F. Codd) permite reemplazar paso a paso, las relaciones entre datos de tipo jerárquico o en red, a una estructura en forma de tabla bidimensional sin perder ninguna de las relaciones existentes entre los datos. Dichas tablas son matrices rectangulares con las siguientes características:

1. Cada entrada de la tabla representa un elemento de dato. No hay grupos repetitivos.
2. Son homogéneos por columna, es decir, todos los elementos de una columna son de la misma clase.
3. Cada columna tiene nombre propio
4. Todos los renglones son diferentes; no se admiten renglones duplicados

5. Tanto los renglones como las columnas pueden considerarse en cualquier secuencia y en cualquier momento, sin afectar por ello ni el contenido de información ni el significado de cualquier función que utilice la tabla.

### Esquemas de Relación

Para comenzar, se darán a continuación algunas definiciones básicas de los conceptos utilizados en el manejo de Bases de Datos Relacionales:

#### Relación:

Dada una serie de conjuntos  $D_1, D_2, \dots, D_n$  (no necesariamente distintos), se dice que  $R$  es una Relación sobre estos  $n$  conjuntos, si es un conjunto de  $n$  tuplas ordenadas  $\langle d_1, \dots, d_n \rangle$  tales que  $d_1$  pertenece a  $D_1$ ,  $d_2$  pertenece a  $D_2$ , ...,  $d_n$  pertenece a  $D_n$ . Los conjuntos  $D_1, D_2, \dots, D_n$  son los dominios de  $R$ . El valor  $n$  es el grado de  $R$ .

#### Entidad:

Cosa u objeto sobre el cual se almacena información.

#### Elemento de Dato:

Unidad mínima identificable.

#### Campo:

Nombre del atributo o identificador de cada elemento.

#### Atributos:

Cada una de las columnas de la tabla (campos) las cuales contienen una característica de la entidad.

#### Tuplas:

Agrupamiento de elementos de datos o atributos, es decir, los renglones o registros de cada tabla.

**Dominio:**

Conjunto de campos o atributos.

**Clave:**

Atributo o conjunto de atributos utilizados para identificar un registro o tupla.

**Clave Primaria:**

Aquella que se utiliza para definir unívocamente un registro o tupla; esto es, el identificador de entidad formado por uno o más atributos. Esta clave es de gran importancia para la computadora, puesto que se utiliza para localizar al registro o tupla mediante un índice o algoritmo de direccionamiento.

**Clave Secundaria.**

Aquellas que no identifican registros únicos, sino aquellos que tienen cierta propiedad.

### **Bases de Datos Relacionales**

Una base de datos se denomina Base de Datos Relacional cuando está construida a base de matrices planas de elementos de datos (tablas).

La relación o tabla es un conjunto de "tuplas" ( renglones) y campos (columnas) donde el número de columnas corresponde al "grado" de la relación. Las relaciones de grado 2 se llaman binarias, las de grado 3 terciarias y las de grado n enearias.

El conjunto de valores de un mismo tipo, es decir, los elementos de cada relación, se denominan "dominio". La columna j-ésima constituye el dominio j-ésimo de la relación.

Como mencionamos anteriormente, todos los archivos de datos se pueden representar como archivos planos (tabla).

Para empezar, un archivo que sería plano si no existiera en él un determinado grupo repetitivo, sería normalizado inmediatamente removiendo dicho grupo repetitivo y formando con él otra tabla o archivo plano aparte, debidamente construido.

En la Fig. 1.8, las tuplas del nuevo archivo deben estar provistas de claves adecuadas para ser identificadas unívocamente. En el caso de esta figura, #-FACT se repite en el archivo DETALLE, y se le combina con #-ART para formar un identificador exclusivo. Como se observa, de esta forma aumenta en cierto grado la redundancia, ya que el elemento #-FACT aparece 2 veces.

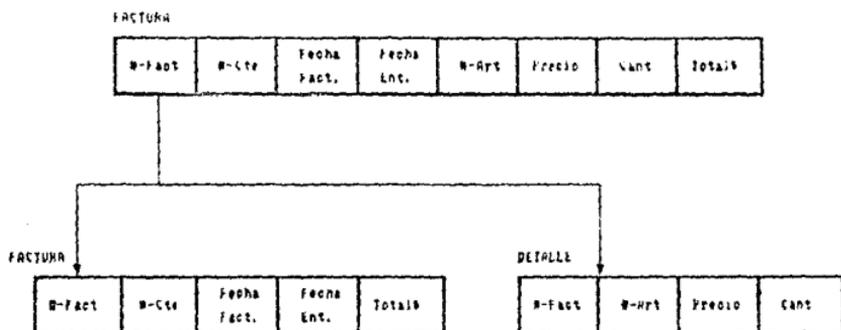


FIG. 1.8 Definición de llaves de registros

Sin embargo, la normalización exige que algunos elementos de datos aparezcan en más de un registro con la finalidad de identificarlos. Esta duplicidad no implica necesariamente un aumento de las necesidades de espacio en almacén, ya que esto va relacionado con las estructuras lógicas (representación de los datos), más que con la forma en que se representan físicamente.

Si los grupos repetitivos no se separan de esta forma, al diseñar inicialmente la base de datos, se corre el riesgo de que la siguiente evolución de ésta exija esa separación con posterioridad, lo cual puede implicar serios cambios en los programas y nuevas pruebas (incremento del costo de mantenimiento).

## Liaves

Toda tupla debe estar asociada con una clave que permita su identificación. A veces la tupla se identifica mediante un solo atributo.

En el ejemplo anterior, #·FACT identifica las tuplas FACTURA. En algunas ocasiones es necesario recurrir a más de un atributo para lograr la identificación inequívoca de una tupla. Este es el caso de la tupla DETALLE, cuya llave deberá estar formada por la fusión de 2 atributos: #·FACT y #·ART. Por lo tanto, se dice que la llave debe cumplir los siguientes requisitos:

1. Identificación unívoca: en cada tupla de una relación, el valor de la clave debe identificar unívocamente esa tupla.
2. No redundancia: ningún atributo de la clave podrá descartarse sin destruir la propiedad de identificación unívoca.

En cada tupla puede existir más de un conjunto de atributos capaz de satisfacer estos 2 requisitos. Estos conjuntos se denominan "llaves candidatas" de las cuales se selecciona una "llave primaria" usada para identificar el registro. Para simplificar la Fig 1.8, tendremos la siguiente notación:

FACTURA: (#·FACT, #·PROV, FECHA-FACT, FECHA-ENT, TOTALS)  
 DETALLE: (#·FACT, #·ART, PRECIO, CANT)

donde el rubro que antecede al paréntesis es el nombre de la relación, los nombres dentro del paréntesis son los dominios y los subrayados corresponden a las llaves primarias.

## Base de Datos Jerárquica

A continuación se verá una forma normalizada de la estructura de árbol que muestra la Fig. 1.3.

DEPTO: (#·DEPTO, NOM-DEP, INF-A, JEFE, PRESUP)  
 PUESTO: (#·DEPTO, #·PUESTO, DESC-PUESTO)  
 EMPLEADO: (#·EMP, NOM-EMP, #·DEPTO, SEXO, SUELDO)  
 ANTEC\_PUESTO: (#·EMP, FECHA, PUESTO)

HIJOS: (#-EMP, NOM-HIJOS, EDAD-HIJOS, SEXO-HIJOS)

NOTA: Como veremos más adelante a estas relaciones deberá aplicarse un paso más de normalización.

### Base de Datos en Red

En la Fig. 1.9 se muestra un esquema en red y una de varias formas normalizadas posibles. Algunas de estas vinculaciones representan relaciones entre datos.

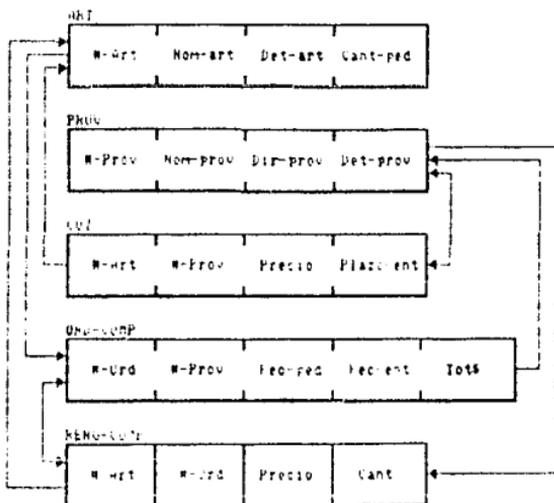


FIG. 1.9 Posibles relaciones entre los registros de una base de datos en red

Como se observa, algunas vinculaciones de la Fig. 1.9 son innecesarias, ya que forman parte de referencias cruzadas que lo único que originan es la saturación del esquema con líneas y flechas.

ART: (#-ART, NOM-ART, DET-ART, CANT-PED)  
 PROV: (#-PROV, NOM-PROV, DIR-PROV, DET-PROV)  
 COT: (#-ART, #-PROV, PRECIO, PLAZO-ENT)  
 ORD-COMP: (#-ORD, #-PROV, FEC-PED, FEC-ENT, TOTS)  
 RENG-COMP (#-ART, #-ORD, PRECIO, CANT)

En general, las líneas con punta de flecha inducen al diseñador de la base de datos lógica a dibujar estructuras inflexibles, engañosas y a veces inválidas. En cambio, la representación relacional es siempre flexible, fácil de comprender y matemáticamente rigurosa si se ejecuta debidamente.

### 1.5 ALGEBRA RELACIONAL Y CALCULO RELACIONAL

Una vez establecidas las diversas tablas, es posible realizar una serie de operaciones tales como unir (JOIN) varias de ellas en una sola, o seleccionar únicamente las tuplas o los dominios deseados de la tabla (SELECT y PROJECT).

Dichas selecciones se realizan por medio de operadores basados en el **Algebra Relacional** y el **Cálculo Relacional**.

Mediante el Algebra Relacional, el usuario declara un conjunto de operaciones de alto nivel que deben ejecutarse sobre ciertas relaciones (u otros agrupamientos de elementos de datos). Estos operadores pueden ser de dos tipos: los operadores relacionales de la teoría de conjuntos como son la unión, intersección, diferencia y producto cartesiano, y los operadores relacionales especiales de selección, proyección, unión y división.

Aún cuando el álgebra es menos "amigable con el usuario" que lenguajes tales como QUE y SQL, es importante debido a que sirve como patrón para medir otros lenguajes. También sienta las bases para la investigación de varios aspectos de la administración de bases de datos, tales como el diseño de bases de datos, la definición de vistas y la reestructuración.

Un modelo de bases de datos relacional consta de dos componentes principales:

1. la estructura de datos relacionales o normalización (ver sección 1.6)
2. el álgebra relacional

Se dice que un sistema de bases de datos es **totalmente relacional** cuando soporta bases de datos relacionales (incluyendo los conceptos de dominio y llave, y las reglas de integridad o normalización), y soporta al menos un lenguaje tan potente como el álgebra relacional.

Un sistema **semirrelacional** es aquel que soporta bases de datos relacionales, pero que tiene un lenguaje menos potente.

El Cálculo Relacional es un cálculo de predicados aplicado, propio de las bases de datos relacionales (una notación que sirve para definir una relación que ha de derivarse de las relaciones ya existentes en la base de datos). Es el más alto nivel de automatización mediante el cual, el usuario manifiesta lo que quiere y el sistema se ocupa de satisfacerlo. Por esta razón es más difícil de implementar que el Álgebra Relacional.

La Tabla 1 muestra algunos símbolos y operadores utilizados en el Cálculo Relacional.

A continuación se presentan algunos ejemplos de la aplicación del Cálculo Relacional basados en las tablas de la Fig. 1.10. Como se puede observar, dichas tablas son de grado 5 (número de dominios o columnas) con cardinalidad 5 (número de tuplas o renglones).

N	Nombre	N	Carrera	Edad
1	J. Lopez	1	10	23
2	M. Lopez	2	5	24
3	L. Gomez	3	3	14
4	M. Gomez	4	4	18
5	R. Lopez	10	7	21

N	Nombre	Duración
6	Matemática	4 años
7	Física	4 años
8	Química	5 años
9	Derecho	5 años
10	Filosofía	5 años

FIG. 1.18 Ejemplos de tablas relacionales

Tabla I. Símbolos del Cálculo Relacional

SÍMBOLOEXPLICACION

x,y

Conjunto de valores de los elementos de datos en el dominio "y" de la relación "x"

A(x,y, x<sub>1</sub>,y<sub>1</sub>,...)Una relación llamada "A" formada por los dominios que contienen los conjuntos de valores x, y, x<sub>1</sub>, y<sub>1</sub>, ...

"De modo que". La expresión a la izquierda de los dos puntos indica que es lo que debe obtenerse, mientras que la expresión escrita a la derecha es un calificador

∃

"Existe"

∀

"Para todo"

"y". Valen a la vez las condiciones unidas con este símbolo

"o". Una cualquiera de las condiciones unidas con este símbolo

"no". No es aplicable la condición escrita a la derecha del símbolo

=, ≠, &lt;, &gt;

Igual a, no igual a, menor que, mayor que

"x"

El valor literal de x

1. Utilizando la relación ALUMNO de la Fig. 1.10, obtener una relación llamada Q, que contenga el conjunto de valores NOMBRE-ALUMNO para todos los alumnos de la carrera número 6:

$Q(\text{ALUMNO.NOMBRE-ALUMNO}) \text{ ALUMNO \#-CARRERA } = 6$

Q

Nombre Alumno
J. Pérez M. Gómez

2. Utilizando la misma relación ALUMNO, obtener una relación que contenga los dominios NOMBRE-ALUMNO y CALIF para cada uno de los alumnos de la carrera 6 con calificación mayor a 8:

$Q(\text{ALUMNO.NOMBRE\_ALUMNO.ALUMNO.CALIF}) \text{ ALUMNO CARRERA } = 6 \wedge \text{ALUMNO CALIF } > 8$

Q

Nombre Alumno	Calif
J. Pérez	10

3. Usando las relaciones ALUMNO y CARRERA, obtener una relación Q en la que aparezca el nombre del alumno, la carrera y la duración para los alumnos que tengan más de 20 años:

$Q(\text{ALUMNO.NOMBRE-ALUMNO.ALUMNO \#-CARRERA, CARRERA DURACION}) \text{ ALUMNO EDAD } > 20$   
 $\wedge \text{ALUMNO \#-CARRERA } = \text{CARRERA \#-CARRERA}$

Q

Nombre Alumno	N Carrera	Duración
J. Pérez	6	4 años
M. López	7	4 años
M. Ortiz	10	3 años

## 1.6. NORMALIZACION

Como se mencionó anteriormente, la normalización es un proceso que le da a una serie de relaciones (tablas) toda la versatilidad y máxima protección posible contra futuros cambios que la base de datos pueda tener.

Ahora se verá cada uno de los pasos necesarios para generar una tabla normalizada.

### Primera Forma Normal (1FN)

*"Una relación R está en "Primera Forma Normal (1FN)" si y sólo si todos los dominios subyacentes solamente contienen valores atómicos".*

Esta primera regla establece que no es posible tener una tabla como la que muestra la Fig. 1.11, ya que el atributo LOCALIDAD no contiene un sólo valor atómico simple, sino uno compuesto.

N-Prov	Non-prov	Localidad		N-Art	Non-art
P1	Berol	Uro.	Uro.	A1	lapices
P1	Berol	Uro.	Uro.	A2	plumas
P1	Berol	Uro.	Uro.	A3	cuadernos
P2	Stilo	Jal.	Guad.	A1	lapices
P2	Stilo	Jal.	Guad.	A2	plumas
P3	hevado	E.C.S.	La Paz	A3	cuadernos

FIG. 1.11 Tabla antes de aplicar la 1FN

La Fig. 1.12 es una representación de la tabla de la Fig. 1.11 después de haber pasado por la 1FN, ya que todos sus elementos son atómicos.

#-Prov	Non-prov	Estado	Ciudad	#-Art	Non-art
P1	Berol	Uro.	Qro.	A1	JAPICES
P1	Berol	Qro.	Qro.	A2	PLUMAS
P1	Berol	Qro.	Qro.	A3	CUADERNOS
P2	Stilo	Jal.	Guad.	A1	JAPICES
P2	Stilo	Jal.	Guad.	A2	PLUMAS
P3	NEVADO	E. U. S.	LA FAZ	A3	CUADERNOS

FIG. 1.12 Tabla después de aplicar la 1FN

Antes de continuar con la Segunda Forma Normal, es necesario reconocer cuáles atributos dependen de otros, para lo cual se tiene la siguiente definición de Dependencia Funcional.

*"Dada una relación R, el atributo Y de R es "Funcionalmente dependiente" del atributo X de R si y sólo si cada valor de X en R tiene asociado exactamente un valor de Y en R (en cualquier instante)"*

Decir que Y es funcionalmente dependiente de X equivale a decir que Y depende de X; esto es, que si en algún momento conocemos el valor de X, el valor de Y quede automáticamente determinado.

Considerando la relación de la Fig. 1.12, tenemos la siguiente estructura:

PEDIDO (#-PROV, NOM-PROV, EDO, CD, #-ART, NOM-ART)

Para observar mejor estas dependencias funcionales, se utilizará el diagrama de la Fig. 1.13. Si Y es funcionalmente dependiente de X, se dibuja una flecha que va de X a Y. (Se toma esta notación para estar de acuerdo a la notación utilizada en lógica matemática, donde  $X \rightarrow Y$  significa que X identifica a Y).

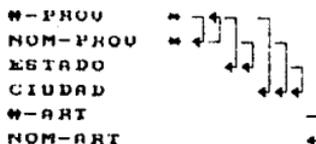


FIG. 1.13 Esquema de relación de la tabla de la fig. 1.12

Los asteriscos distinguen a los atributos primos (miembros de llaves candidatas).

Como se observa, un mismo atributo puede ser funcionalmente dependiente de un grupo de atributos más que de uno solo.

A continuación se tiene la siguiente definición de **Dependencia Funcional Completa**:

*"Un atributo o colección de atributos Y, de una relación R, es "Dependiente funcional completo" de otra colección de atributos X de la misma relación R, si Y es funcionalmente dependiente del total de X pero no de ningún subconjunto de X".*

Con esta definición se puede observar como el atributo ESTADO, por ejemplo, es dependiente funcional completo de la llave concatenada #PROV y NOM\_PROV, es decir, de los atributos primos.

### Segunda Forma Normal (2FN)

*"Una relación R está en Segunda Forma Normal (2FN) si y sólo si está en 1FN y cada uno de sus atributos no primos es dependiente funcional completo de cada llave primaria de R."*

Esto implica que todos los atributos no primos deben depender forzosamente de la llave primaria, no importando si dependen de otras llaves secundarias u otros atributos no primos.

Como se dijo anteriormente, un atributo es no primo si no participa en la llave primaria.

La Fig. 1.14 representa la 2FN de la tabla de la Fig. 1.12.

#-PROV	NOM-PROV	ESTADO	CIUDAD
P1	Berol	Uro.	Uro.
P1	Berol	Uro.	Uro.
P1	Berol	Uro.	Uro.
P2	Stilo	Jal.	Guad.
P2	Stilo	Jal.	Guad.
P3	Nezaco	Erlab.	La Paz

#-ART	NOM-ART
A1	lapices
A2	plumas
A3	cuadernos
A1	lapices
A2	plumas
A3	cuadernos

FIG. 1.14 Tablas después de aplicar la 2FN

Como se observa, fué necesario descomponer esta tabla en 2 sub-tablas, ya que NOM\_ART no dependía de #-PROV sino únicamente de #ART.

En el diagrama de la Fig. 1.15 se observa la dependencia de los atributos no primos con las llaves primarias.

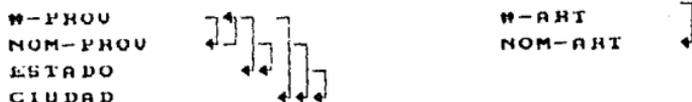


FIG. 1.15 Esquemas de relación de las tablas de la fig. 1.14

### Tercera Forma Normal (3FN)

*"Una relación R está en Tercera Forma Normal (3FN) si y sólo si está en 2FN, y todos y cada uno de sus atributos no primos son dependientes no transitivos de la llave primaria de R"*

Esto significa que todos los atributos no primos pueden depender única y exclusivamente de la llave primaria. En este caso, no se permite la dependencia hacia cualquier otro atributo que nos sea esta misma llave.

Los diagramas de la Fig. 1.16 muestran las tablas de la Fig 1.14 en su 3FN.

N-PROV	NOM-PROV	ESTADO	CIUDAD	N-ART	NOM-ART	RELACION		
P1	Herol	Qro.	Qro.	A1	lapices	P1	Qro.	A1
P2	Stilo	Jal.	Guad.	A2	plumas	P1	Qro.	A2
P3	Nevado	B.C.S.	La Paz	A3	cuadernos	P2	Jal.	A1
						P4	Jal.	A2
						P3	B.C.S.	A3

FIG. 1.16 Tablas relacionales después de aplicar la 3FN

Como se observa, fué necesario descomponer la relación en 4 tablas, donde cada una de ellas ha eliminado las dependencias transitorias de las tablas anteriores. También han sido eliminadas aquellas tuplas que estaban duplicadas.

El diagrama de la Fig. 1.17 muestra cómo quedaron estas relaciones después de haber aplicado la 3FN.



FIG. 1.17 Esquemas de relación de las tablas de la fig. 1.16

Para concluir este capítulo, se detallarán las ventajas que reporta la representación de datos en la forma normalizada, y particularmente en la 3FN.

**NOTA:** Aún cuando en este trabajo se hace referencia hasta la 3FN, es importante señalar que existen otras formas normales, las cuales van agregando una serie de restricciones al universo de las bases de datos.

### Ventajas del Uso de la 3FN

1. **Facilidad de uso.** La manera más fácil de representar la mayor parte de los datos para el usuario inexperto, es la que se basa en el empleo de tablas bidimensionales.
2. **Flexibilidad.** Las operaciones del tipo Proyección y Unión permiten partir y juntar relaciones con el fin de dar al usuario los archivos que necesita para sus aplicaciones en el formato en que lo requiera.
3. **Precisión.** Los enlaces dirigidos, muy comunes en las representaciones lógicas actuales, son frecuentemente engañosos. Las Relaciones (tablas) tienen un significado preciso y pueden ser manipuladas mediante el Álgebra o el Cálculo Relacional.
4. **Seguridad.** Con este modelo es más fácil implementar los controles de seguridad, ya que las autorizaciones de seguridad se refieren a las relaciones y van ligadas a los atributos más sensibles.
5. **Relacionabilidad.** Se tiene la máxima flexibilidad para relacionar atributos de diferentes conjuntos de tuplas o archivos (tablas).
6. **Facilidad de implementación.** El almacenamiento físico de los archivos planos es eventualmente menos complejo que el de los árboles o las estructuras en red.
7. **Independencia de datos.** Las bases de datos están obligadas a crecer a causa de la adición de nuevos atributos, nuevas tuplas y nuevas relaciones. Se les usará también de distintas maneras. Si la base de datos está normalizada con independencia de datos en el software, será más sencillo reestructurar los datos sin tener que volver a escribir los programas de aplicación. De igual forma, esto reducirá o controlará los costos de mantenimiento.
8. **Lenguaje para la manipulación de datos.** El lenguaje utilizado en la manipulación de los datos podrá estar basado en el álgebra o cálculo relacional, lo cual es una forma de homogeneizar estos lenguajes.
9. **Claridad.** La representación de los datos en tablas facilita su comprensión y su uso, no así en el caso de las estructuras jerárquicas o en red, donde el diseño de las gráficas utilizando líneas y flechas lo hace menos legible.

## CAPITULO 2: ANALISIS DE LOS REQUERIMIENTOS DE INFORMACION DEL DEPARTAMENTO DE SERVICIO SOCIAL

Este capítulo comienza con la descripción de la técnica de análisis estructurado de sistemas desarrollada por Gane y Sarson, la cual se basa en la definición de Diagramas de Flujo de Datos (DFD), y en la especificación de cada uno de los elementos que componen el sistema analizado. Posteriormente se muestra la utilización de esta técnica para realizar el análisis del sistema de información que operará en el Departamento de Servicio Social.

### 2.1 TECNICA DE ANALISIS ESTRUCTURADO DE SISTEMAS

Como se mencionó anteriormente, este capítulo está basado en el empleo de la técnica de análisis estructurado de sistemas en la cual se desarrolla un Diagrama Lógico de Flujo de Datos (DFD). Este diagrama utiliza los cuatro símbolos que se muestran en la Fig. 2.1, los cuales permiten diseñar modelos lógicos que representen el comportamiento de los procesos de información que se están analizando.

Las entidades externas son generalmente clases lógicas de cosas o personas, las cuales representan una fuente o destino de transacciones. Cada entidad externa se encuentra fuera de los límites del sistema que se está considerando (ver Fig. 2.2). Un flujo de datos está constituido por cualquier medio por el cual los datos pasan de una entidad o proceso a otro. Un proceso es un grupo de actividades manuales o automatizadas (o ambas) que transforman el contenido del flujo de información. El almacenamiento de datos es el lugar donde se encuentran depositados los datos que utilizan los procesos. También se conocen como archivos.

Un DFD puede hacerse tan detallado como se desee (dependiendo del nivel de análisis requerido) utilizando estos cuatro símbolos.

La Fig. 2.2 muestra un DFD completo donde se han señalado los posibles límites de automatización, es decir, los procedimientos administrativos que deberán realizarse manualmente (P4) y los procesos computacionales (P3, P5, P6, P7, P8, P9 y P10).



FIG. 2.1 Símbolos del D.F.D.

Además del diseño del DFD, es posible implementar otras herramientas de análisis de sistemas, como es la elaboración de un diccionario de datos, el cual se utiliza para identificar cada uno de los elementos de datos que se encuentran presentes en el flujo de datos, darles nombres significativos, definir cada uno de ellos y organizarlos de tal manera que su definición sea fácilmente localizable. El uso de árboles y tablas de decisión y del lenguaje estructurado o el pseudocódigo, permiten graficar la estructura lógica del sistema y expresar sus políticas en una forma global y sin ambigüedad. Por último, los diagramas de acceso inmediato permiten diseñar la forma más eficaz de tener acceso a los datos que se encuentran en los archivos del sistema (ver Fig. 2.3).

## 2.2 ANÁLISIS DEL FLUJO DE DATOS DEL DEPARTAMENTO DE SERVICIO SOCIAL

A continuación se analiza en forma detallada el flujo de datos que manejan los procesos realizados por el Departamento de Servicio Social, determinando así las partes principales en las cuales intervendrá el Sistema de Control de Trámites y Evaluación del Servicio Social (SICTESS).

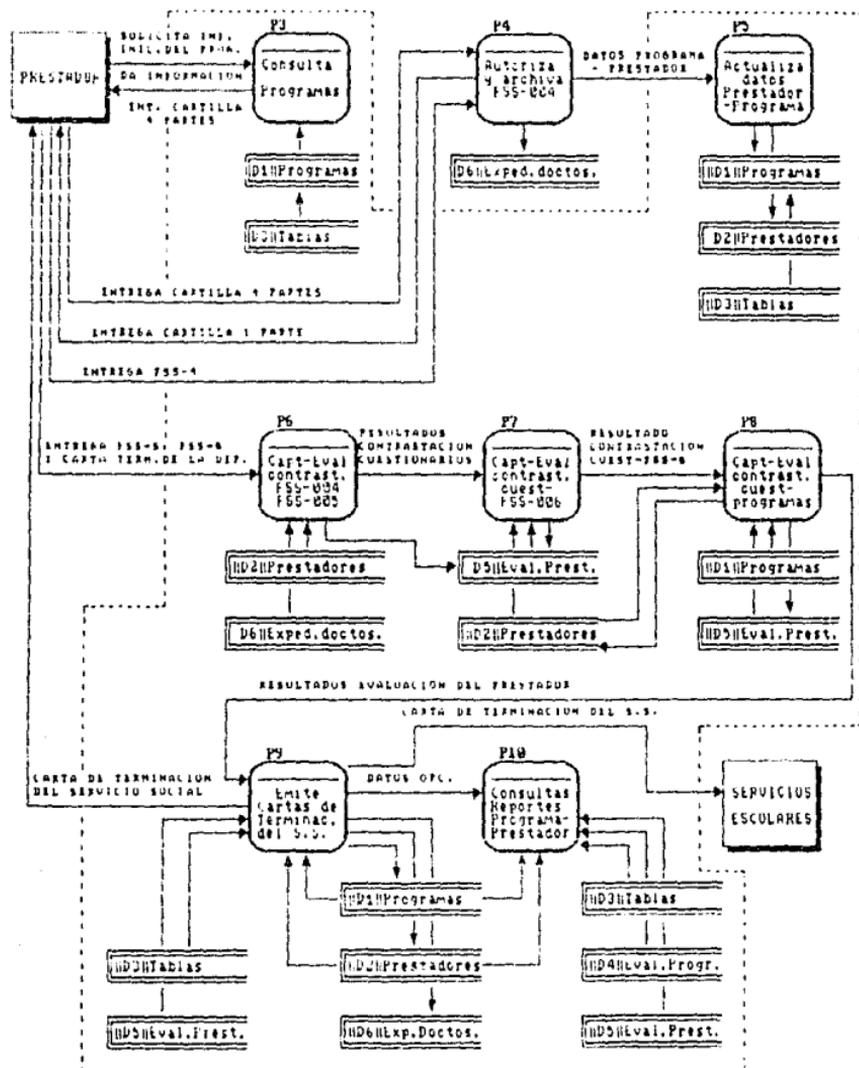


FIG. 2.2 Posibles límites de automatización

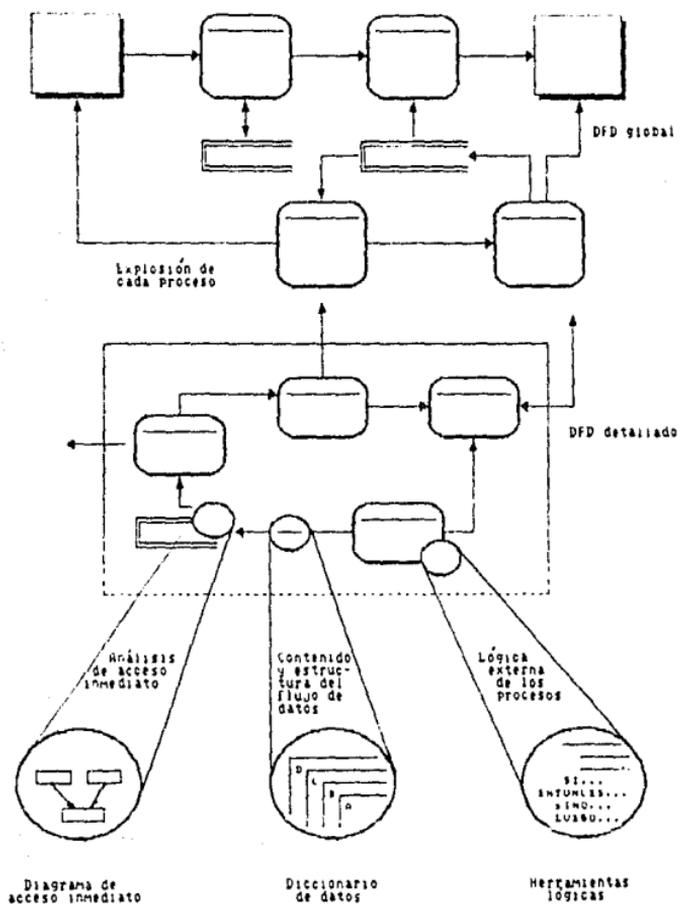


FIG. 2.3 Componentes de un modelo lógico

Para llevar a cabo esta actividad, se realizó en primera instancia la revisión de los registros manejados por el departamento, así como la observación de las actividades ahí ejecutadas, lo cual permitió extraer la información suficiente.

### **Identificación de los Estados Administrativos del**

#### **Departamento de Servicio Social**

El Departamento de Servicio Social de la Escuela Nacional de Estudios Profesionales Acatlán está compuesto por las siguientes secciones:

1. Una Jefatura de Departamento.

Esta unidad tiene como objetivo el procurar el óptimo funcionamiento del Departamento, coordinando, organizando, supervisando y autorizando todas las actividades y documentos que se requieran en el mismo. La información generada por el SICTESS servirá como herramienta de apoyo para la toma de decisiones de esta sección.

2. Una Sección de Planeación, Evaluación y Servicio Social Profesional.

El objetivo de esta sección es el optimizar las actividades del departamento, la prestación y los programas de Servicio Social y Bolsa de Trabajo, e implementar la realización del Servicio Social Profesional. Aquí es donde el SICTESS juega uno de sus papeles más importantes, ya que apoyará los procesos de evaluación tanto de los programas como de los prestadores, agilizándolos y reportando información muy valiosa sobre el desempeño y aprovechamiento del Servicio Social.

3. Una sección de Promoción y Difusión.

El objetivo de esta sección es el obtener una óptima promoción y difusión de las actividades del departamento, de los programas y de la prestación del Servicio Social y de la Bolsa de Trabajo. Esta es la sección encargada de captar, registrar y difundir los programas a la comunidad estudiantil, la cual se verá apoyada mediante el SICTESS al obtener ágilmente toda la información de los programas requerida por los prestadores.

4. Una sección de Trámites Internos y Externos.

Esta sección tiene como objetivo el procurar la óptima tramitación de la documentación correspondiente al Servicio Social. Entre sus funciones se encuentran las de tramitar, registrar, procesar, archivar y depurar toda la documentación manejada por el departamento. Como apoyo a esta sección, el SICTESS facilitará todo el registro y depuración de información de los prestadores, generando los informes requeridos por el prestador y las demás secciones del D.S.S.

5. Una sección de Bolsa de Trabajo.

El objetivo esencial de esta sección es el contribuir a la satisfacción de las necesidades de empleo profesional de los solicitantes y las del mercado de trabajo profesional, buscando una mayor identificación entre ambas, en función de la formación profesional. La agilización y optimización de los procesos ejecutados por esta sección, no se encuentran contemplados en este trabajo, por lo que serán objeto de algún trabajo posterior.

Además de las múltiples interrelaciones que hay entre cada una de estas secciones, existen otras entidades externas que también interactúan con el Departamento de Servicio Social, como son:

- El Prestador (estudiante o egresado) que realiza o realizará su Servicio Social
- El Solicitante (estudiante o egresado) que solicita los servicios de la Bolsa de Trabajo (no incluido en este trabajo)
- La Comisión Coordinadora del Servicio Social en Acatlán, la cual regula las actividades del D.S.S.
- La Dependencia o institución donde se realizará el Servicio Social
- La Empresa o institución que requiere del trabajo profesional (no incluido en este trabajo)
- El Departamento de Servicios Escolares, el cual es el encargado de captar las cartas de terminación del servicio social, para llevar el control de la historia académica del estudiante

Los siguientes diagramas representan las interrelaciones que existen entre el Departamento de Servicio Social y las entidades externas, así como las relaciones que existen entre las mismas secciones que componen al departamento.

En el diagrama de la Fig. 2.4 se detalla cada una de las actividades que se realizan actualmente entre el Departamento de Servicio Social y las diferentes entidades externas descritas anteriormente.

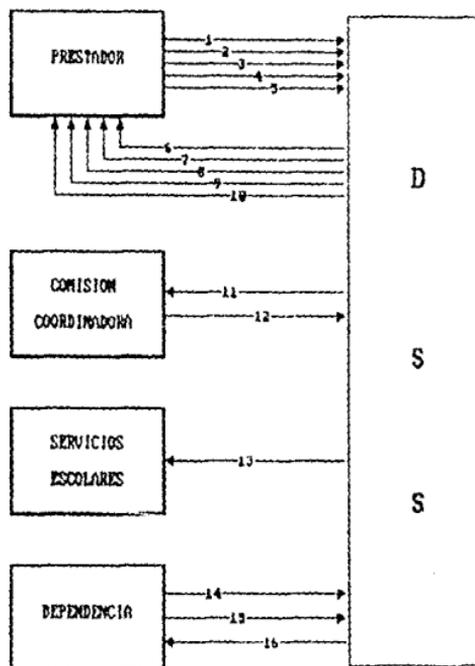


FIG. 2.4 Diagrama de flujo de información (original)

**NOTACION:** El primer número de la lista es la referencia utilizada en la elaboración del diagrama, y el número que se encuentra entre paréntesis al final de cada línea, indica el orden en que se realiza cada una de las actividades en la organización.

1. Solicita información (3)
2. Entrega documentos (5)
3. Entrega Cartilla autorizada por la Dependencia (8)
4. Entrega FSS-004 lleno (12)
5. Entrega Cartilla, FSS-005 y FSS-006 llenos y carta de terminación de la dependencia (14)
6. Informa de los programas y la documentación inicial requerida (4)
7. Entrega Cartilla (4 partes) y comprobante de documentos (6)
8. Entrega Cartilla (2 partes) y FSS-004 para llenar (11)
9. Entrega comprobante de documentos, FSS-005 y FSS-006 para llenar (13)
10. Entrega carta de terminación del servicio social (15)
  
11. Entrega Cartilla (4 partes) para autorización (9)
12. Entrega Cartilla autorizada (3 partes) (10)
  
13. Entrega carta de terminación del servicio social (16)
  
14. Aviso de rechazo/aceptación del prestador (7)
15. Propone programa de servicio social (1)
16. Aviso de rechazo/aceptación del programa (2)

En el diagrama de la Fig. 2.5 se detallan las actividades que se van a hacer más eficientes una vez que el sistema haya sido implementado.

1. Entrega Cartilla y cuestionarios llenos (4)
2. Entrega Cartilla (4 partes) y FSS-003 para llenar (3)
3. Entrega carta de terminación del servicio social (5)
  
4. Entrega carta de terminación del servicio social (6)
  
5. Propone programa de servicio social (1)
6. Emite aviso de rechazo/aceptación del programa (2)

El diagrama de la Fig. 2.6 representa la interrelación que existe entre cada una de las secciones que constituyen actualmente al Departamento de Servicio Social, es decir, describe la estructura interna del departamento.

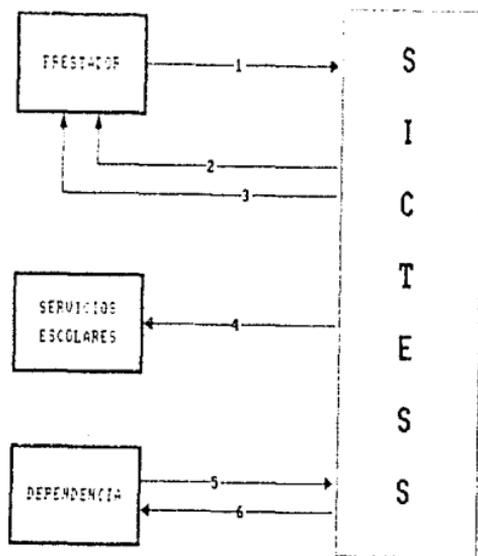


FIG. 2.5 Diagrama de Flujo de Información (revisado)

1. Recibe documentos para autorización (4) (14)
2. Entrega documentos autorizados (Cartilla, evaluaciones y carta de terminación del servicio social) (5) (15)
3. Recibe FSS-004 (19)
4. Recibe FSS-005 y FSS-006 (22)
5. Recibe programas de servicio social (2)
6. Aviso de rechazo o aceptación del prestador (24)
7. Aviso de rechazo o aceptación del programa (3)
8. Capta propuestas de servicio social de las dependencias (1)
9. Recibe aviso de programas aceptados (6)
10. Publica los programas aceptados (7)
11. Asigna clave a los programas de servicio social (8)
12. Solicita detalles del programa (10)
13. Recibe documentos (12)
14. Recibe Cartillas rechazadas por la dependencia (16)
15. Recibe Cartillas autorizadas inicia el servicio social (20)
16. Recibe carta de terminación de la dependencia (21)
17. Recibe aviso de rechazo-aceptación del prestador (17)

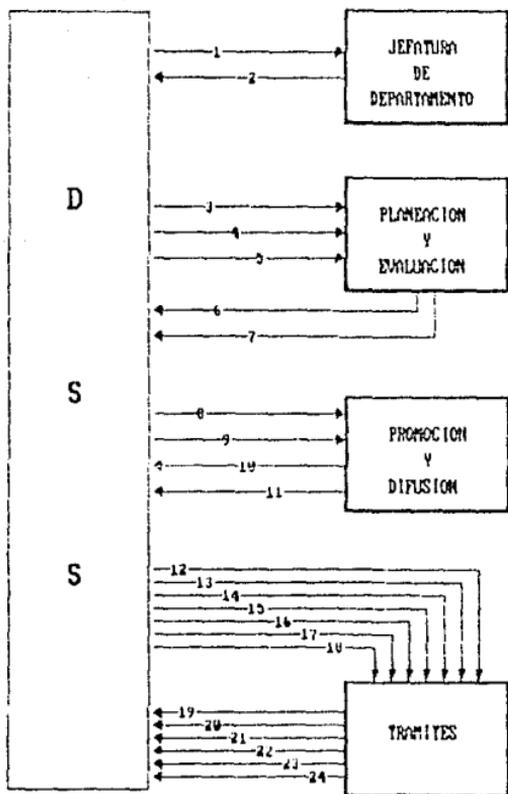


FIG. 2.6 Diagrama de flujo de información (original)

18. Recibe aviso de rechazo-aceptación del programa (9)
19. Presta información detallada de los programas (11)
20. Devuelve Cartilla (13)
21. Asigna número de expediente (18)
22. Entrega cartas de terminación de prestadores aceptados (25)
23. Envía programas para evaluaciones subsecuentes (26)
24. Turna programa evaluado, FSS-004, FSS-005 e informe final (23)

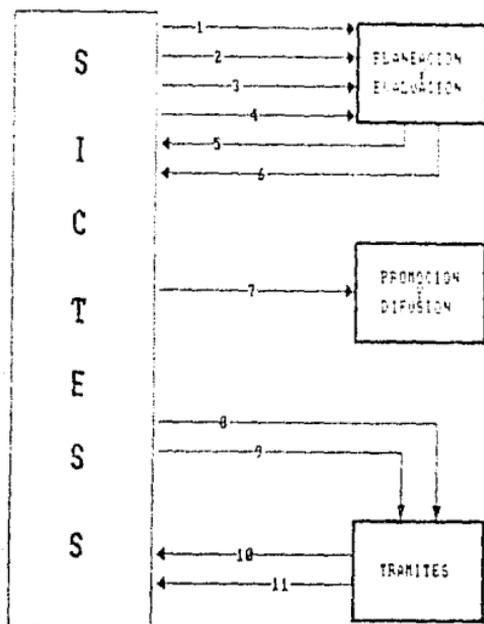


FIG. 2.7 Diagrama de flujo de información (revisado)

El diagrama de la Fig. 2.7 detalla cada una de las actividades que se realizarán entre las distintas secciones del departamento, una vez que este haya sido automatizado.

1. Registra la evaluación del programa (2)
2. Registra la contrastación de cuestionarios (6)
3. Registra la contrastación de cuestionarios-informe global (7)
4. Registra la contrastación de cuestionarios, informe global-programa de servicio social (8)
5. Emite aviso de rechazo-aceptación del programa (3)
6. Emite aviso de rechazo-aceptación del prestador (9)
7. Registra los programas de servicio social (1)
8. Registra Cartillas aceptadas-rechazadas (5)
9. Registra la terminación del servicio social (10)
10. Consulta los programas de servicio social (4)
11. Emite cartas de terminación del servicio social (11)

### 2.3 DESARROLLO DE LOS DIAGRAMAS DE FLUJO DE DATOS (DFD)

A continuación se presentan los diagramas de flujo de datos tanto de los procesos de evaluación de los programas de servicio social, como los procesos de trámites y evaluación de los prestadores.

El diagrama de la Fig. 2.8 muestra en forma general el proceso de captación y evaluación de los programas de servicio social.

En este diagrama se tiene una Entidad externa, que es la dependencia que propone el programa de servicio social, dos procesos P1 y P2 de captación de programas y asignación de claves y evaluación del servicio social, dos archivos de datos D1 y D3 que contienen los datos de los programas de servicio social y todos los parámetros que se utilizan a lo largo del sistema (carreras, tipos de programas, ubicaciones, áreas y subáreas, etc.) y dos archivos D4 y D6, los cuales contienen los resultados de las evaluaciones.

A continuación se presenta la descripción general de los elementos que integran este diagrama.

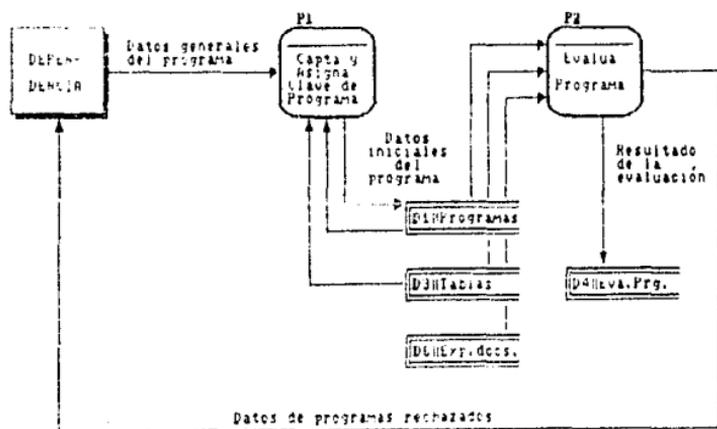


FIG. 2.8 DFD de la Evaluación de los programas de Servicio Social

<u>No.</u>	<u>NOMBRE</u>	<u>FUNCION.</u>
P1	Captación y asignación de clave del programa	Este proceso es el encargado de registrar los datos generales de cada uno de los programas captados. Asignándoles una clave de identificación. Esta clave permitirá dar seguimiento al programa durante su utilización en el sistema.
P2	Evaluación del programa	Este proceso realiza todas las evaluaciones de los programas que sean necesarias. Una vez que el programa ha sido evaluado y aceptado, todos sus datos son archivados en el sistema para su control posterior.
D1	Programas	Archivo que contiene todos los datos del programa de servicio social.
D3	Tablas	Archivo que contiene los parámetros que permitirán clasificar a los programas y prestadores del servicio social, como las carreras, los tipos de programa, las áreas y subáreas de ubicación, etc.
D4	Eval. prog.	Archivo que contiene los resultados de la evaluación del programa.
D6	Exp. doctos.	Este es un archivo físico que constituye el expediente que contiene los documentos correspondientes a cada prestador.

El diagrama de la Fig. 2.9 muestra cada uno de los elementos que intervienen en el proceso en sí de trámites y evaluación de los prestadores del servicio social.

Como se puede observar, existen dos Entidades externas, que son el prestador y el departamento de servicios escolares; ocho procesos diferentes del P3 al P10, que serán descritos posteriormente, y seis tipos de archivos diferentes, del D1 al D6, que contienen todos los datos relativos a los programas y prestadores del servicio social.

A continuación se tiene la descripción de cada uno de las partes procesos que integran este diagrama.

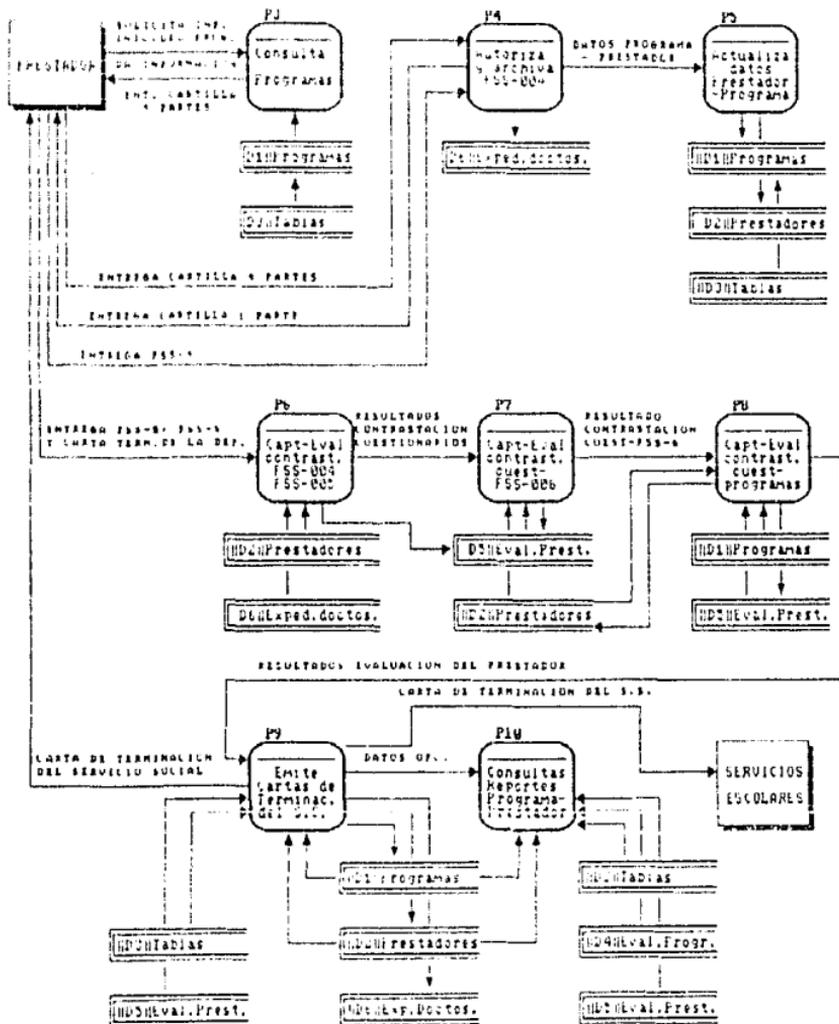


FIG. 2.9 D.F.D. de los trámites y evaluación de los prestadores de Servicio Social

<u>No.</u>	<u>NOMBRE</u>	<u>FUNCION</u>
P3	Consulta de programas	Este proceso permitirá obtener toda la información que se requiera sobre algún programa específico. La consulta se realizará a partir del número de identificación del programa.
P4	Autorización y archivo del FSS-004	Este es un proceso manual, el cual se encarga de autorizar el inicio del servicio social, y archivar el cuestionario inicial (FSS-004). Este cuestionario será utilizado posteriormente para realizar la evaluación del prestador.
P5	Actualiza los datos del prestador	Este proceso permite actualizar los datos del prestador y mantener su registro en forma definitiva en el sistema.
P6	Captura de la contrastación del FSS-004 y el FSS-005	En este proceso se realiza el registro de la contrastación del cuestionario inicial (FSS-004) y el final (FSS-005) obteniendo el resultado respectivo.
P7	Captura de la contrastación cuestionarios-FSS-006	En este proceso se realiza el registro de la contrastación anterior con el informe global de actividades (FSS-006), obteniendo el resultado respectivo.
P8	Captura de la contrastación cuestionarios programa	En este proceso se lleva a cabo el registro de la contrastación anterior con el programa del servicio social, generando así el resultado final de la evaluación del prestador.
P9	Emisión de cartas de terminación del servicio social	Este proceso genera las cartas de terminación a partir del resultado de la evaluación del prestador.
P10	Consultas y reportes de los programas y prestadores del servicio social	Este proceso constituye el generador de pantallas de consulta y de reportes estadísticos del sistema.
D1	Programas	Archivo que contiene todos los datos del programa de servicio social.

D2	Prestadores	Archivo que contiene todos los datos de los prestadores del servicio social.
D3	Tablas	Archivo que contiene los parámetros que permitirán clasificar a los programas y prestadores del servicio social como las carreras, los tipos de programa, las áreas y subareas de ubicación, etc.
D4	Eval. prog.	Archivo que contiene los resultados de la evaluación del programa.
D5	Eval. prest.	Archivo que contiene los resultados de la evaluación del prestador.
D6	Exp. doctos.	Este es un archivo físico (folder) que constituye el expediente que contiene los documentos correspondientes a cada prestador.

Por último se detallan los elementos que componen cada proceso antes mencionado indicando cuales son los datos de entrada y cuál es la información que genera.

### PROCESO 1: CAPTACION Y ASIGNACION DE LA CLAVE DEL PROGRAMA

<u>ENTRADAS</u>	<u>PROCESOS</u>	<u>SALIDAS</u>
- Programas de Servicio Social	P.1.1 Capta los datos del programa	
	P.1.2 Verifica la existencia del programa. Si existe, genera aviso y rechaza el programa  Si no existe, asigna clave del programa y registra sus datos, verificando que los campos tipo, canal, area, subarea, actividad, apoyo, carreras y semestre existan en el sistema	Mensaje "el programa ya existe"  D1: Programas

## PROCESO 2: EVALUACION DEL PROGRAMA

<u>ENTRADAS</u>	<u>PROCESOS</u>	<u>SALIDAS</u>
- D1: prog. - D3: tablas # programa - Contrasta- ción del 1er plano: correspon- dencia interna del programa - # programa - Contrasta- ción del 1er plano: correspon- dencia programa- concepto del servicio social	P.2.1 Registra las respuestas positivas de la 1a. con- trastación	D4: eval.progs. Result.1a.contr
- D4: eval. progs.	P.2.2 Registra las respuestas positivas de la 2a. con- trastación	D4: eval.progs. Result.2a.contr
- D4: eval. progs.	P.2.3 Obtiene los resultados de la evaluación del 1er plano	D4: eval.progs. Fecha Resul.1er plano Genera reporte y consulta del resultado
- Contrasta- ción del 2do plano: conceptos- percepción del servicio social	P.2.4 Registra las respuestas positivas de la evalua- ción del 2do plano	Arch. eval progs Fecha Resul.2do plano
- D4: eval. progs.	P.2.5 Obtiene los resultados de la evaluación del 2do. plano	Genera reporte y consulta del resultado

P.2.6	Actualiza el status del programa	D4: eval.progs. Genera reporte y consulta del status
-------	----------------------------------	---

### PROCESO 3: CONSULTA DE PROGRAMAS

<u>ENTRADAS</u>	<u>PROCESOS</u>	<u>SALIDAS</u>
# programa - D1: progs. - D3: Tablas	P.3.1 Captura del número del programa	Información de los datos del programa

### PROCESO 4: AUTORIZACION Y ARCHIVO DE FSS-004

<u>ENTRADAS</u>	<u>PROCESOS</u>	<u>SALIDAS</u>
- Cartilla - FSS-004	P.4.1 Proceso manual encargado de recolectar las cartillas autorizadas	D6: exp.doctos

### PROCESO 5: ACTUALIZA DATOS DEL PRESTADOR

<u>ENTRADAS</u>	<u>PROCESOS</u>	<u>SALIDAS</u>
- D1: progs. - D2: prest. - D3: Tablas	P.5.1 Se complementan los datos del prestador, una vez que su servicio social ha sido autorizado	D2:Prestador
	P.5.2 Se actualiza la solicitud del prestador en el archivo de programas	D1:Programas

### PROCESO 6: CAPTURA DE LA CONTRASTACION DE CUESTIONARIOS FSS-004 Y FSS-005

<u>ENTRADAS</u>	<u>PROCESOS</u>	<u>SALIDAS</u>
- D2: prest. - D4: exp.doc. (FSS-004) - FSS-005	P.8.1 Registra la contrastación entre los cuestionarios FSS-004 y FSS-005	D5: eva.prest. Result.1a.contr

### PROCESO 7: CAPTURA DE LA CONTRASTACION DE CUESTIONARIOS-FSS-006

<u>ENTRADAS</u>	<u>PROCESOS</u>	<u>SALIDAS</u>
- D2: prest. - D5: eval. prest. (1a.contr) - FSS-006	P.7.1 Registra la contrastación entre los cuestionarios (1a.contrastación) y el informe global (FSS-006)	D5: eva.prest. Result.2a.contr

### PROCESO 8: CAPTURA DE LA CONTRASTACION DE CUESTIONARIOS Y EL PROGRAMA

<u>ENTRADAS</u>	<u>PROCESOS</u>	<u>SALIDAS</u>
- D2: prest. - D5: eval. prest. (2a.contr) - D1: progr. - D5: eval. prest. (1a.,2a., 3a. contr)	P.8.1 Registra la contrastación entre los cuestionarios (2a.contrastación) y el programa de servicio social	D5: eval.prest Result.3a.contr
	P.8.2 Obtiene el resultado final de la evaluación del prestador, actualizando su status	D5: eval. prest

### PROCESO 9: EMITE CARTAS DE TERMINACION DEL SERVICIO SOCIAL

<u>ENTRADAS</u>	<u>PROCESOS</u>	<u>SALIDAS</u>
- D5: eval. prest. (final)	P.9.1 Genera cartas de terminación de aquellos prestadores que hayan realizado satisfactoriamente su servicio social	D1: Programa
- D1: progr.		D2: Prestador (emisión de la carta)
- D2: prest.		D6: exp. docs.
- D3: tablas		

### PROCESO 10: CONSULTAS Y REPORTES DE LOS PROGRAMAS Y PRESTADORES

<u>ENTRADAS</u>	<u>PROCESOS</u>	<u>SALIDAS</u>
- D1: progr.	P10.1 Genera las pantallas de captura y los reportes estadísticos que requieren los usuarios del sistema	Reportes y consultas
- D2: prest.		
- D3: tablas		
- D4 eval. progr.		
- D5: eval. prest.		

Además de estos procesos de intercambio de información entre el D.S.S. y las entidades externas, existen otros procesos internos (realizados dentro del departamento) que complementarán el mantenimiento del sistema y la explotación de la información. Estos procesos son:

- Mantenimiento a las tablas del sistema, donde se tienen los diferentes tipos de programas, canales de funcionamiento, carácter, áreas y subareas de ubicación, actividades, apoyos del programa, carreras y semestres solicitados.
- Procesos eventuales y anuales que depuren la información del periodo, dejando únicamente los datos relevantes. Estos procesos eliminarán los programas y prestadores que no fueron aceptados. También eliminará toda aquella información del periodo que no sea útil exclusivamente para efectos estadísticos.

## CAPITULO 3: DISEÑO DEL SISTEMA

En este capítulo se realiza el diseño del sistema utilizando una estructura descendente (top-down) y modular. Posteriormente se elabora el diseño de la base de datos que será implementada.

### 3.1 DISEÑO DESCENDENTE (TOP-DOWN) DE LA ESTRUCTURA DEL SISTEMA

#### 0 MENU GENERAL

Este menú constituye la entrada al sistema (SICTESS). Controla todos los procedimientos de captura de datos, así como los procesos y generación de consultas y reportes de que dispone el sistema. (Fig. 3.1)

#### 1 MENU DE CAPTURA

Este menú controla la entrada al sistema de los datos de los programas de servicio social y de los prestadores, así como la captura de las tablas que contienen los parámetros del mismo. (Fig. 3.2)

#### 1.1 MENU DE TABLAS

Este menú controla la captura e impresión de las tablas del sistema.

#### 1.1.1 CAPTURA DE TABLAS

Este procedimiento permite realizar la actualización (altas, bajas y cambios) y validación de las tablas con los parámetros del sistema.

#### 1.1.1.1 al 1.1.1.9

Selección de cada uno de los parámetros del sistema que se deseen actualizar.

#### 1.1.2 IMPRESION DE TABLAS

Este procedimiento permite realizar la impresión de las tablas del sistema.

#### 1.1.2.1 al 1.1.2.9

Selección de cada uno de los parámetros del sistema que se deseen imprimir.

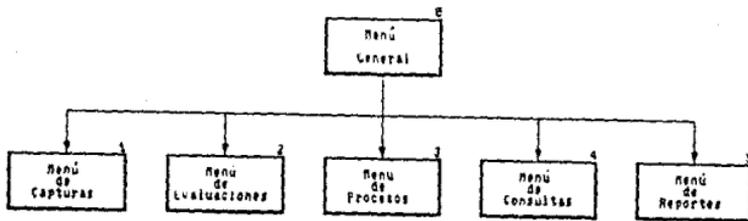


FIG. 3.1 Menú general

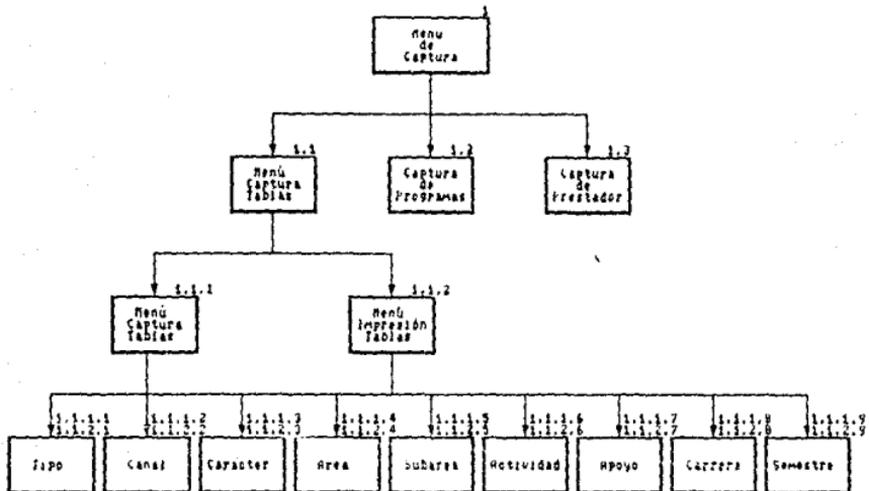


FIG. 3.2 Menú de captura

## 1.2 CAPTURA DE PROGRAMAS

Este procedimiento permite realizar la actualización y validación de los datos de los programas que están disponibles para el departamento.

## 1.3 CAPTURA DE PRESTADORES

Este procedimiento permite realizar la actualización y validación de los datos de los prestadores de servicio social.

## 2 MENU DE EVALUACION

Este menú controla los procesos de evaluación de programas y prestadores de servicio social. (Fig. 3.3)

### 2.1 MENU DE EVALUACION DE PROGRAMAS

Este menú controla los procesos de captura, evaluación y consulta de los programas de servicio social

#### 2.1.1 CAPTURA DE LA EVALUACION DEL PRIMER PLANO

Este procedimiento permite registrar los resultados de la contrastación del primer plano del programa de servicio social.

#### 2.1.2 PROCESO DE EVALUACIÓN DEL PRIMER PLANO

Este procedimiento calcula el resultado de la contrastación del primer plano.

#### 2.1.3 CAPTURA DE LA EVALUACION DEL SEGUNDO PLANO

Este procedimiento permite registrar los resultados de la contrastación del segundo plano del programa de servicio social.

#### 2.1.4 PROCESO DE EVALUACION DEL SEGUNDO PLANO

Este procedimiento calcula el resultado de la contrastación del segundo plano.

#### 2.1.5 CONSULTA DEL STATUS DEL PROGRAMA

Este procedimiento despliega en pantalla los resultados de la evaluación del programa.

#### 2.1.6 EMISION DEL STATUS DEL PROGRAMA

Este procedimiento genera los resultados impresos de la evaluación del programa.

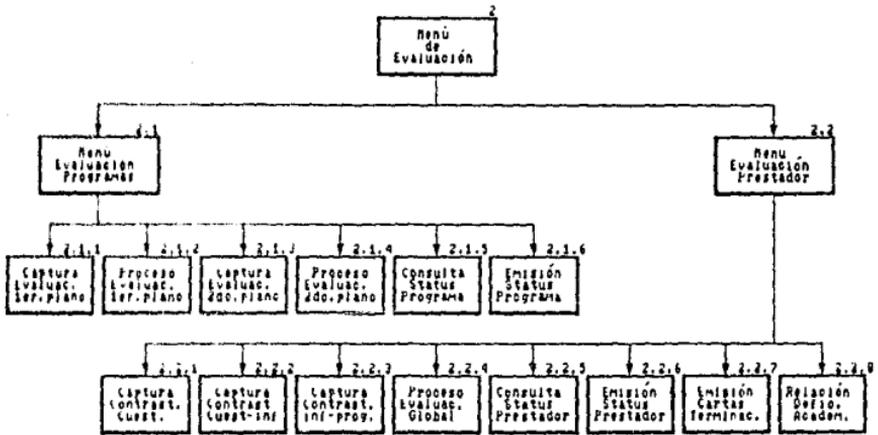


FIG. 3.3 Menú de evaluación

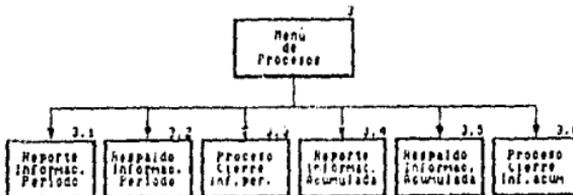


FIG. 3.4 Menú de Procesos

### 3 MENU DE PROCESOS

Este menú controla los procesos de depuración del sistema. (Fig. 3.4)

#### 3.1 REPORTE DE INFORMACION DEL PERIODO

Este procedimiento permite generar un listado del detalle de la información estadística que se encuentra en el sistema perteneciente a un período determinado (cálculo de la estadística y emisión del listado).

#### 3.2 RESPALDO DE LA INFORMACION DEL PERIODO

Este procedimiento permite realizar el respaldo de la información que se encuentra en un período determinado, tal como la relación de los programas que ya han sido cubiertos o de los prestadores que ya han terminado su servicio social

#### 3.3 PROCESO DE CIERRE DE INFORMACION DEL PERIODO

Este procedimiento elimina el detalle de la información de un período específico, misma que ha sido previamente listada o respaldada en diskette, quedando disponible únicamente la información estadística a que dió origen.

#### 3.4 RESPALDO DE INFORMACION ACUMULADA

Este procedimiento genera un listado de la información estadística que tiene acumulada el sistema hasta la fecha: un año, un semestre, etc., (cálculo de la estadística y emisión del listado)

#### 3.5 RESPALDO DE LA INFORMACION ACUMULADA

Este procedimiento permite respaldar la información estadística acumulada, tal como el número de prestadores que concluyeron su servicio social, o el número de prestadores que realizaron su servicio social en determinada area y subarea de ubicación.

#### 3.6 PROCESO DE CIERRE DE INFORMACION ACUMULADA

Este procedimiento elimina la información estadística acumulada, misma que ha sido previamente listada o respaldada en diskette. Una vez realizado este proceso, la información de los archivos queda eliminada para comenzar un nuevo conteo.

### 4 MENU DE CONSULTA

Este menú controla las diferentes pantallas de consulta de información de que dispone el sistema. (Fig. 3.5)

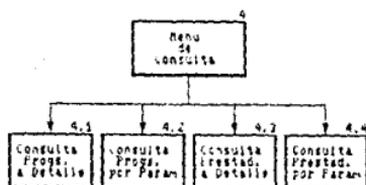


FIG. 3.5 Menú de consulta

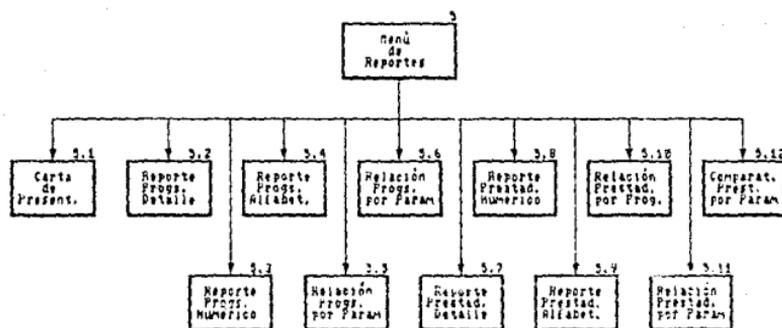


FIG. 3.6 Menú de reportes

#### 4.1 CONSULTA DE PROGRAMAS A DETALLE

Este procedimiento permite consultar toda la información correspondiente a un programa en particular.

#### 4.2 CONSULTA DE PROGRAMAS POR PARAMETRO

Este procedimiento obtiene la relación de todos los programas que están asignados a un parámetro en particular.

#### 4.3 CONSULTA DE PRESTADORES A DETALLE

Este programa permite consultar toda la información correspondiente a un prestador en particular.

#### 4.4 CONSULTA DE PRESTADORES POR PARAMETRO

Este procedimiento obtiene la relación de los prestadores que están asignados a un parámetro determinado.

#### 5 MENU DE REPORTES

Este menú controla la generación de los diferentes reportes estadísticos correspondientes a los programas y prestadores del servicio social.

#### 5.1 CARTA DE PRESENTACION

Este procedimiento genera la impresión de las cartas de presentación de los prestadores que lo soliciten.

#### 5.2 REPORTE DE PROGRAMAS A DETALLE

Este procedimiento genera un listado con toda la información correspondiente a un programa en particular (ver punto 4.1).

#### 5.3 REPORTE DE PROGRAMAS NUMERICO

Este procedimiento genera la relación de programas de servicio social por número de programa.

#### 5.4 REPORTE DE PROGRAMAS ALFABETICO

Este procedimiento genera la relación de programas de servicio social por nombre.

#### 5.5 RELACION DE PROGRAMAS POR PARAMETRO

Este procedimiento genera la relación de todos los programas de servicio social clasificados por cada uno de los parámetros seleccionados (ver punto 4.2).

**5.6 COMPARATIVO DE PROGRAMAS POR PARAMETRO**

Este procedimiento genera un cuadro comparativo de la relación de los programas pertenecientes a dos parámetros seleccionados.

**5.7 REPORTE DE PRESTADORES A DETALLE**

Este procedimiento genera un listado con toda la información correspondiente a un prestador en particular (ver punto 4.3).

**5.8 REPORTE DE PRESTADORES NUMERICO**

Este procedimiento genera la relación de prestadores de servicio social por número de expediente

**5.9 REPORTE DE PRESTADORES ALFABETICO**

Este procedimiento genera la relación de prestadores de servicio social en orden alfabético.

**5.10 RELACION DE PRESTADORES POR PARAMETRO**

Este procedimiento genera una relación de los prestadores que están inscritos a cada programa de servicio social.

**5.11 RELACION DE PRESTADORES POR PARAMETRO**

Este procedimiento genera la relación de todos los prestadores de servicio social clasificados por cada parámetro (ver punto 4.4).

**5.12 COMPARATIVO DE PRESTADORES POR PROGRAMA**

Este procedimiento genera un cuadro comparativo de la relación de prestadores pertenecientes a dos parámetros seleccionados.

**3.2 DISEÑO DE LA BASE DE DATOS****CAT\_PRG1 (D1): Archivo principal de programas de servicio social**

<u>Nombre del Campo</u>	<u>Tipo de Campo</u>	<u>Long.</u>	<u>Descripción</u>
1. CLA_PRG	N Numérico	6	Clave del programa

2. NOM_PRG	Alfabético	40	Nombre del programa
3. DEP_PRG	Alfabético	40	Nombre de la dependencia que solicita al prestador
4. TIPO	Numérico	2	Clave del tipo de programa
5. CANAL	Numérico	2	Clave del canal de funcionamiento del programa
6. CARACTER	Numérico	2	Clave del carácter del programa
7. AREA	Numérico	2	Clave del área de ubicación del programa
8. SUBAREA	Numérico	2	Clave de la subárea de ubicación del programa
9. ACTIVIDAD	Numérico	2	Clave de la actividad del programa
10. APOYO	Numérico	2	Clave del apoyo del programa
11. CARRERA1	Numérico	2	Clave de la primera carrera solicitada por el programa
12. CARRERA2	Numérico	2	Clave de la segunda carrera solicitada por el programa
13. CARRERA3	Numérico	2	Clave de la tercera carrera solicitada por el programa
14. CARRERA4	Numérico	2	Clave de la cuarta carrera solicitada por el programa
15. CARRERA5	Numérico	2	Clave de la quinta carrera solicitada por el programa
16. CARRERA6	Numérico	2	Clave de la sexta carrera solicitada por el programa
17. CARRERA7	Numérico	2	Clave de la séptima carrera solicitada por el programa
18. CARRERA8	Numérico	2	Clave de la octava carrera solicitada por el programa

19. CARRERA9	Numérico	2	Clave de la novena carrera solicitada por el programa
20. CARRERA0	Numérico	2	Clave de la décima carrera solicitada por el programa
21. SEMESTRE	Numérico	2	Clave del semestre solicitado por el programa
22. STA_PRG	Numérico	1	Estado del programa, dependiendo de la evaluación (aceptado, pendiente o rechazado)

#### **CAT\_PRG2: Archivo secundario de programas de servicio social**

1. CLA_PRG	Numérico	6	Clave del programa
2. UNI_RES	Alfabético	25	Nombre de la unidad responsable del programa de servicio social
3. RES_DEP	Alfabético	25	Nombre del responsable del serviciosocial en la dependencia
4. DOM_DEP	Alfabético	25	Domicilio de la dependencia
5. TEL_DFP	Alfabético	25	Teléfono de la dependencia
6. RES_PRG	Alfabético	25	Nombre del programa de servicio social en la dependencia
7. PTO_PRG	Alfabético	25	Puesto del responsable del servicio social en la dependencia
8. UBI_PRE	Alfabético	25	Domicilio del prestador en la dependencia
9. TEL_PRE	Alfabético	25	Teléfono del prestador en la dependencia
10. FEC_INI	Fecha	8	Fecha de inicio del servicio social
11. FEC_FIN	Fecha	8	Fecha de terminación del servicio social
12. OBJ_PRG	Alfabético	25	Objetivo del programa de servicio social

13. PLA_SOL	Numérico	2	Número total de plazas solicitadas
14. PLA_DIS	Numérico	2	Número de plazas disponibles
15. HOR_PRG	Alfabético	25	Horario requerido
16. REQ_PRG	Alfabético	25	Requisitos del programa
17. OBS_PRG	Alfabético	25	Observaciones sobre el programa

#### **CAT\_PRES (D2): Archivo maestro de prestadores del servicio social**

1. CTA_ALU	Numérico	8	Número de cuenta del prestador
2. NOM_ALU	Alfabético	40	Nombre del prestador
3. CAR_ALU	Numérico	2	Clave de la carrera a la que pertenece el prestador
4. SEM_ALU	Numérico	2	Clave del semestre al que pertenece el prestador
5. DOM_ALU	Alfabético	40	Domicilio del prestador
6. TEL_ALU	Alfabético	25	Teléfono del prestador
7. STA_ALU	Numérico	1	Estado del prestador, dependiendo de la evaluación (aceptado, pendiente o rechazado)
8. CLA_PRG	Numérico	6	Clave del programa al que pertenece el prestador
9. EXP_ALU	Numérico	10	Número de expediente del prestador

#### **TAB\_CAR (D3): Tabla de carreras**

1. CLA_CAR	Numérico	2	Clave de la carrera
2. NOM_CAR	Alfabético	40	Nombre de la carrera

**TAB\_SEM: Tabla de semestres de inscripción al servicio social**

- |            |            |    |                     |
|------------|------------|----|---------------------|
| 1. CLA_SEM | Numérico   | 2  | Clave del semestre  |
| 2. NOM_SEM | Alfabético | 40 | Nombre del semestre |

**TAB\_TIPO: Tabla de tipos de programas**

- |             |            |    |                            |
|-------------|------------|----|----------------------------|
| 1. CLA_TIPO | Numérico   | 2  | Clave del tipo de programa |
| 2. NOM_TIPO | Alfabético | 40 | Nombre del tipo            |

**TAB\_CANAL: Tabla de canal de funcionamiento de los programas**

- |              |            |    |                                   |
|--------------|------------|----|-----------------------------------|
| 1. CLA_CANAL | Numérico   | 2  | Clave del canal de funcionamiento |
| 2. NOM_CANAL | Alfabético | 40 | Nombre del canal                  |

**TAB\_CARAC: Tabla de caracter del programa**

- |              |            |    |                                 |
|--------------|------------|----|---------------------------------|
| 1. CLA_CARAC | Numérico   | 2  | Clave del carácter del programa |
| 2. NOM_CARAC | Alfabético | 40 | Nombre del carácter             |

**TAB\_AREA: Tabla de areas de ubicación del programa**

- |             |            |    |                             |
|-------------|------------|----|-----------------------------|
| 1. CLA_AREA | Numérico   | 2  | Clave del área de ubicación |
| 2. NOM_AREA | Alfabético | 40 | Nombre del área             |

**TAB\_SUBA: Tabla de subareas de ubicación del programa**

1. CLA_SUBAR	Numérico	2	Clave de la subarea de ubicación
2. NOM_SUBAR	Alfabético	40	Nombre de la subarea

**TAB\_ACTI: Tabla de actividades del programa**

1. CLA_ACTIV	Numérico	2	Clave de la actividad del programa
2. NOM_ACTIV	Alfabético	40	Nombre de la actividad

**TAB\_APOY: Tabla de apoyos del programa**

1. CLA_APOYO	Numérico	2	Clave del apoyo del programa
2. NOM_APOYO	Alfabético	40	Nombre del apoyo

**EVA\_PRG (D4): Evaluación del programa**

1. CLA_PRG	Numérico	6	Clave del programa
2. PRI_PRG	Numérico	3.2	Resultado de la primera evaluación
3. FEC1_PRG	Fecha	8	Fecha de la primera evaluación
4. SEG_PRG	Numérico	3.2	Resultado de la segunda evaluación
5. FEC2_PRG	Fecha	8	Fecha de la segunda evaluación
6. EVA_PRG	Numérico	3.2	Resultado final de la evaluación

**EVA\_PRE (D5): Evaluación del prestador**

1. EXP_ALU	Numérico	10	Número de expediente del alumno
2. PRI_ALU	Numérico	3.2	Resultado de la primera contrastación
3. SEG_ALU	Numérico	3.2	Resultado de la segunda contrastación
4. TER_ALU	Numérico	3.2	Resultado de la tercera contrastación
5. EVA_ALU	Numérico	3.2	Resultado de la evaluación del alumno
6. FEVA_ALU	Fecha	8	Fecha de evaluación
7. EVA_MAT	Alfabético	10	Area de mejoramiento académico
8. EVA_ARE	Alfabético	10	Materia de mejoramiento académico
9. EVA_SEM	Numérico	2	Semestre de mejoramiento académico

## CAPITULO 4: DESARROLLO DEL SISTEMA

Este capítulo presenta el desarrollo de los programas correspondientes al SICTESS utilizando el Lenguaje de Flujograma.

### 4.1 INTRODUCCION AL LENGUAJE DE FLUJOGRAMA

El Lenguaje de Flujograma (desarrollado por S.Chapa) es un lenguaje icónico basado en el concepto de flujo de datos, en el cual se ensamblan figuras geométricas con argumentos variables y resultados que determinan implícitamente el control del flujo. Los iconos son abstracciones que se traducen en macroinstrucciones que actúan sobre los objetos para producir una salida.

El propósito de utilizar el lenguaje de flujograma es el de crear procesos que transformen estos iconos mediante un programa ensamblador, en código ejecutable para la solución del problema.

A continuación se da una breve explicación de los iconos que serán utilizados para desarrollar los programas que componen el sistema (fig. 4.1).

- a) **Lectura de Objeto.**  
Indica la relación a partir de la cual se va a procesar información (archivo de entrada).
- b) **Escritura de Objeto.**  
Indica la relación en la cual va a depositarse el resultado obtenido (archivo de salida).
- c) **Proyección.**  
Selecciona los atributos especificados en el ícono, los cuales podrán ser utilizados en algún proceso posterior.
- d) **Junta Natural.**  
Se refiere a la junta (Join) de dos archivos relacionados en base a un atributo en común.

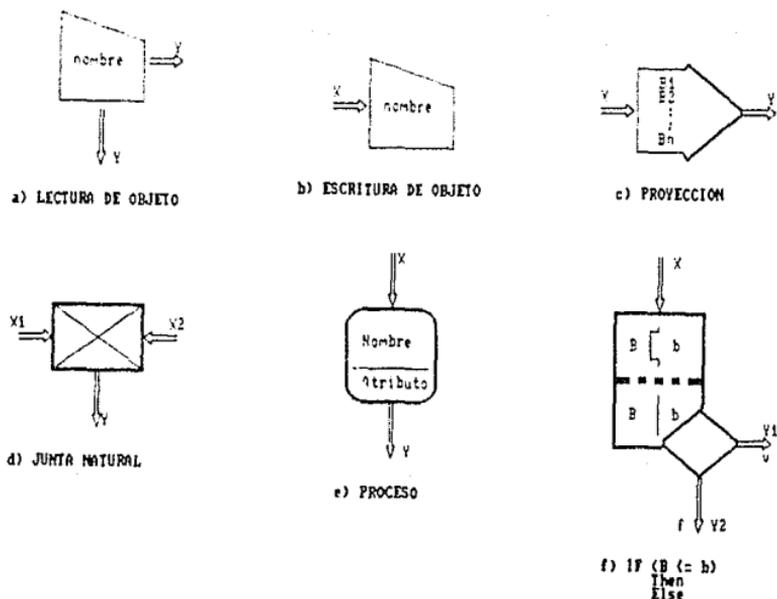


FIG. 4.1 Iconos de Flujoograma

- e) Proceso.  
Este ícono permite especificar el tipo de operación que debe realizarse con los atributos seleccionados.
- f) Condición.  
Permite seleccionar una serie de operaciones a realizar a partir de la construcción de una condición específica.

## 4.2 DESARROLLO DE LOS PROGRAMAS DEL SISTEMA

En esta sección se presenta el desarrollo de los programas que conforman el SICTESS utilizando el lenguaje de flujoograma descrito anteriormente.

Cada uno de los flujoogramas siguientes corresponde a los procesos definidos en los diagramas de flujo de datos del capítulo 2.

La Fig. 4.2 muestra el desarrollo del Flujoograma 1 correspondiente al proceso de Captura de Programas de servicio social. Dicho proceso fué definido por vez primera como proceso P1 (ver fig. 2.8).

Este flujoograma comienza realizando una junta natural de los archivos Programas y Tablas, lo cual permite efectuar posteriormente la captura de los programas de servicio social. Debido a la complejidad de los controles de validación que intervienen en esta captura, fué necesario desarrollar dichos programas directamente en código en dBase III Plus (ver Apéndice).

Una vez que los datos han sido validados, se realiza una proyección de dichos atributos para ser posteriormente almacenados en el archivo de Programas.

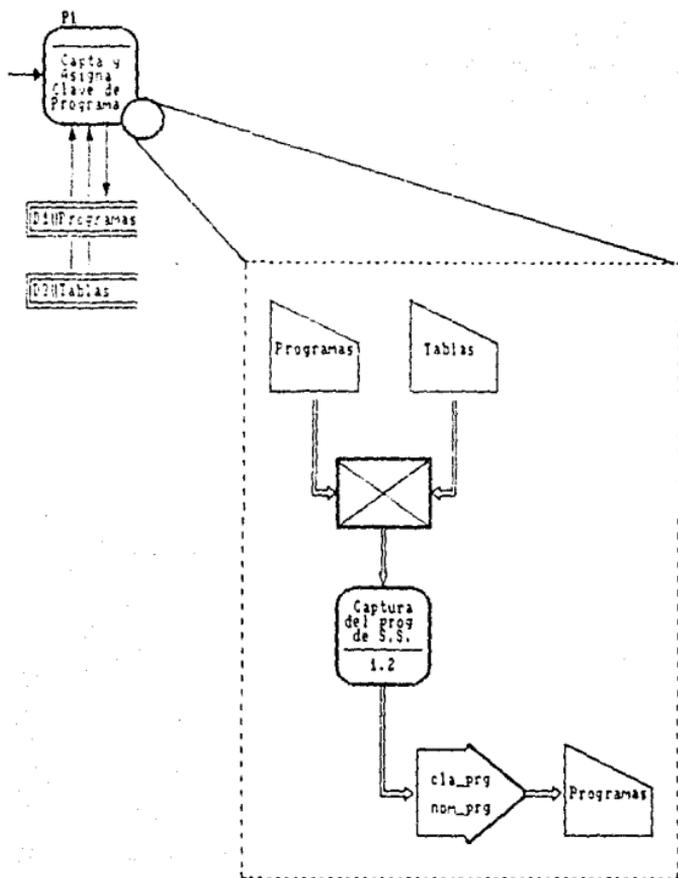


FIG. 4.2 Flujoograma 1: Captura de los programas

La Fig. 4.3 muestra el desarrollo del Flujograma 2 correspondiente al proceso de Evaluación de los Programas de servicio social (ver proceso P2, fig. 2.8).

Para comenzar se realiza una junta natural de los archivos Programas y Tablas. Con el resultado obtenido se realiza una nueva junta pero ahora con el archivo Expediente de Documentos.

A continuación, se realiza el proceso de Contratación del Primer Plano de los Programas de Servicio Social, y posteriormente la Contratación del Segundo Plano. Dichos procesos permitirán decidir si los programas evaluados son aceptados por el Departamento o bien rechazados y devueltos a la dependencia de procedencia.

Una vez obtenido el resultado de la evaluación, se realiza una proyección de los atributos principales que se obtuvieron en la evaluación, para ser almacenados finalmente en el archivo de Evaluación de Programas.

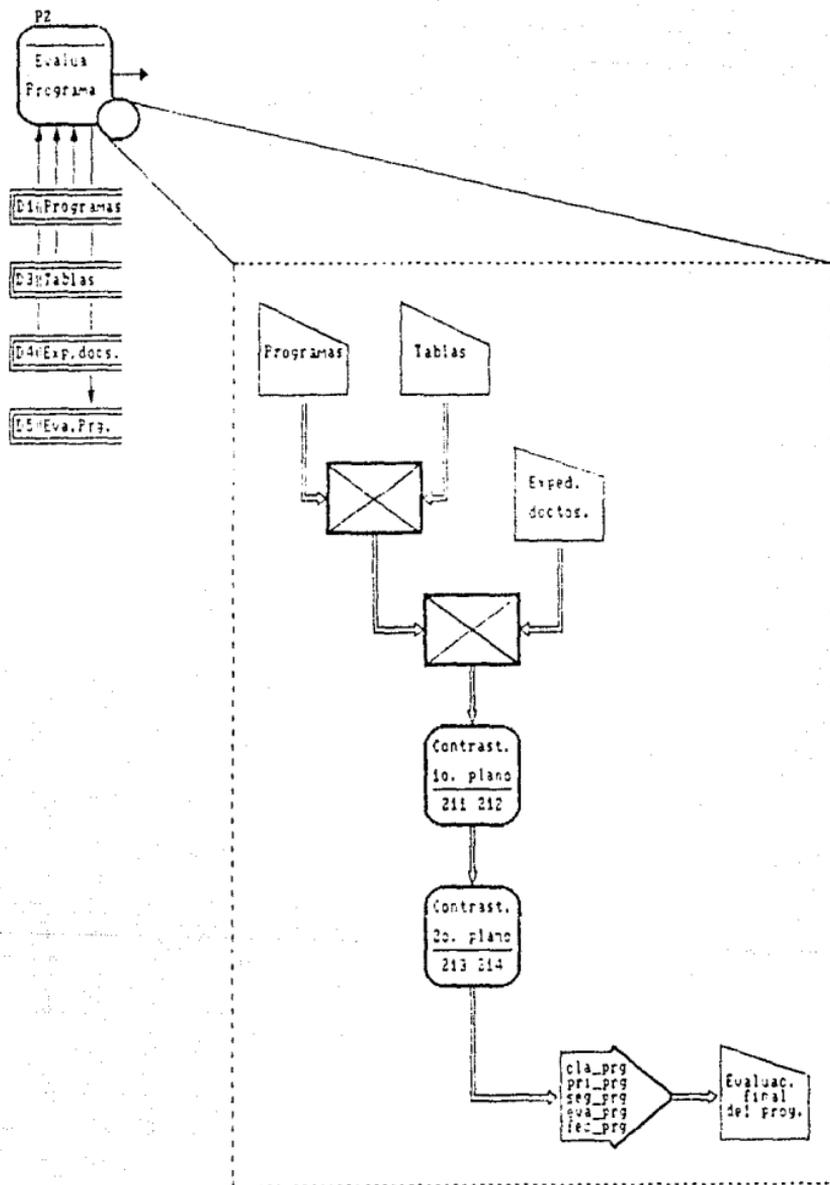


FIG. 4.3 Flujograma 2: Evaluación del programa

En la Fig. 4.4 se muestra el desarrollo del Flujograma 3 correspondiente al proceso de Consulta de los Programas de servicio social (ver proceso P3, fig. 2.9).

El flujograma comienza con una junta natural de los archivos Programas y Tablas.

Posteriormente se realiza el proceso de Consulta de los Programas, el cual está basado principalmente en una serie de proyecciones que presentan en pantalla los atributos requeridos.

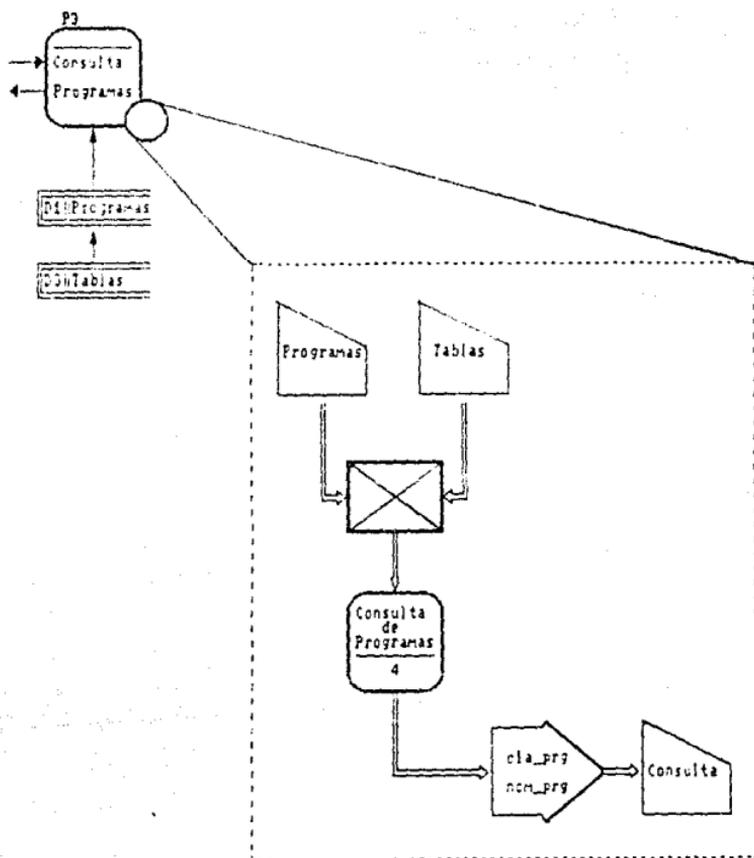


FIG. 4.4 Flujograma 3: Consulta de programas

El Flujograma 4 de la Fig. 4.5 corresponde al proceso de Captura y Actualización de los Prestadores del servicio social (ver proceso P5, fig. 2.9).

Este flujograma comienza con una junta natural de los archivos Programas y Tablas, cuyo resultado se vuelve a juntar con el archivo Prestador.

A continuación se realiza el proceso de Captura de los datos del Prestador que, al igual que la captura de los programas de servicio social, se desarrollaron directamente en dBase III Plus.

Por último se realiza una proyección de los atributos capturados y validados, y su almacenamiento en el archivo Prestador.

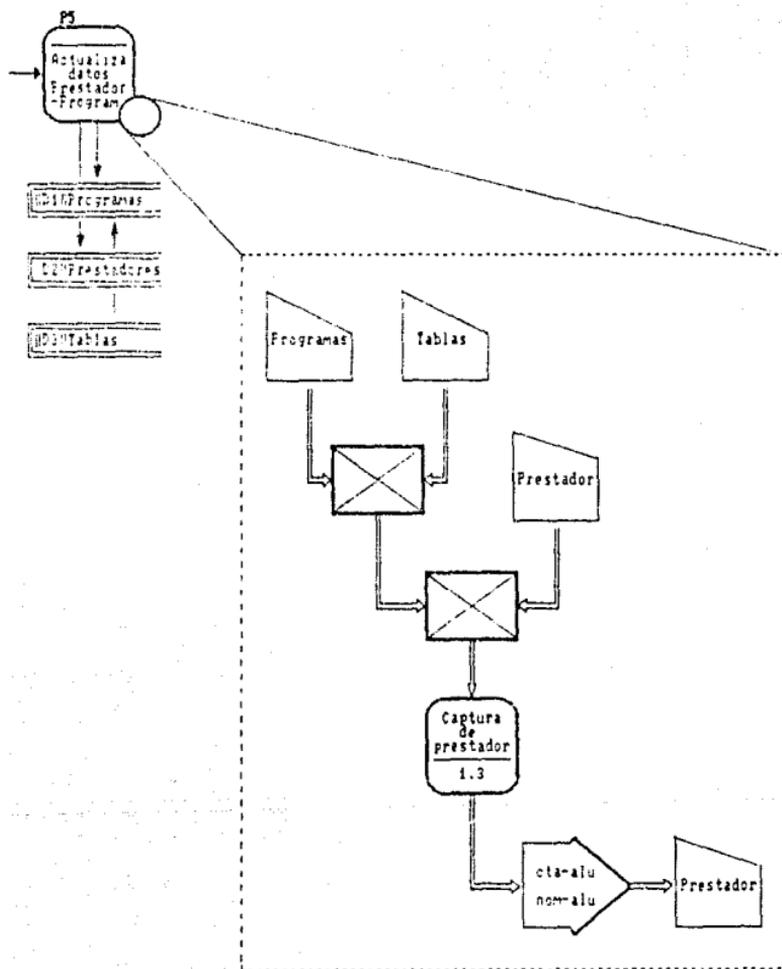


FIG. 4.5 Flujo de datos 4: Captura de prestadores

La Fig. 4.6 muestra el Flujograma 5 correspondiente a la Captura de la Primera Contrastación del Prestador, o Contrastación de los Cuestionarios Inicial y Final FSS-004 y FSS-005, respectivamente (proceso P6, fig. 2.9).

En primera instancia se realiza una junta natural de los archivos Prestador y Expediente de Documentos.

A continuación se lleva a cabo el proceso de Captura de la Primera Contrastación del Prestador (variables A, B y C las cuales corresponden a valores que van de 0 a 10 utilizados para evaluar la correspondencia entre las respuestas que el prestador dió en los cuestionarios FSS-004 y FSS-005), junto con lo cual se efectúan las operaciones que se muestran en el cuadro de "Detalle". Esto permitirá asignar un valor específico (R1) a dicha contrastación.

Para terminar este flujograma, se realiza una proyección de los resultados obtenidos, los cuales se almacenan en el archivo de Evaluación de Prestadores.

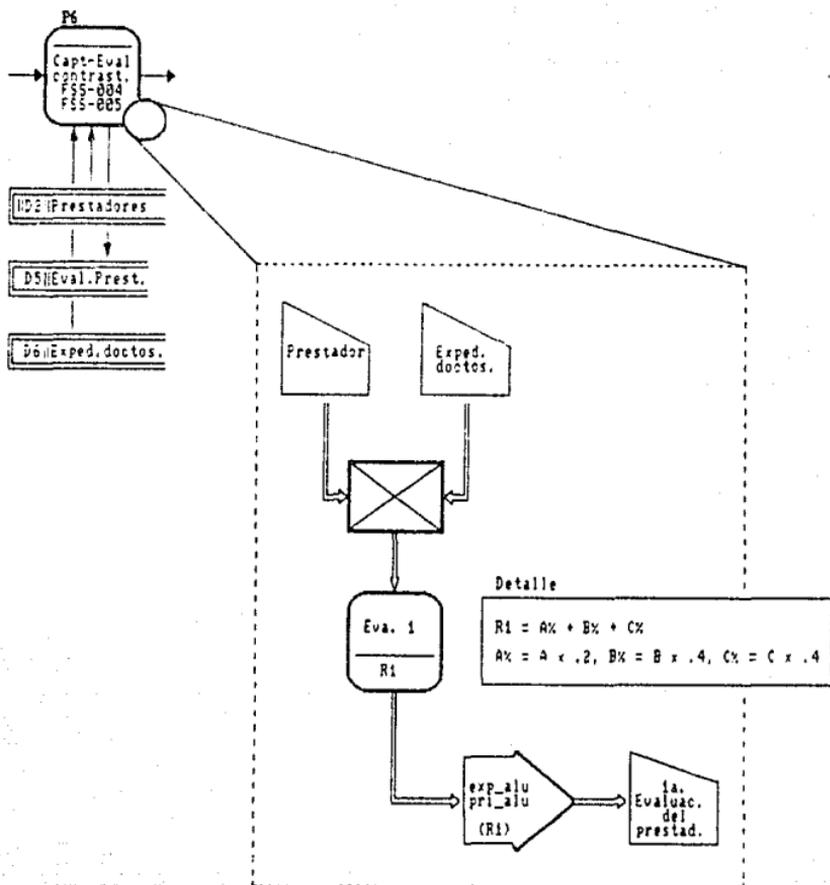


FIG. 4.6 Flujoograma 5: Primera contrastación del prestador

En la Fig. 4.7 se presenta el Flujograma 6, el cual corresponde al proceso de Captura de la Segunda Contrastación del Prestador, o Contrastación de Cuestionarios e Informe Global FSS- 006 (ver proceso P7, fig. 2.9).

El flujograma comienza con una junta natural entre los archivos Prestador y Evaluación del Prestador, para proceder con la Captura de la Segunda Contrastación (variables A, B, C, D, E, F, G, H, I, J, K, L, M, N y O, las cuales corresponden a los valores que van del 0 al 10 utilizados para evaluar la correspondencia entre la contrastación obtenida en el flujograma anterior y las respuestas que dió el prestador en su Informe Global). A continuación se realizarán las operaciones que se encuentran especificadas en el cuadro de "detalle".

Por último se proyecta el resultado obtenido (R2) en un archivo de Evaluación de Prestadores.

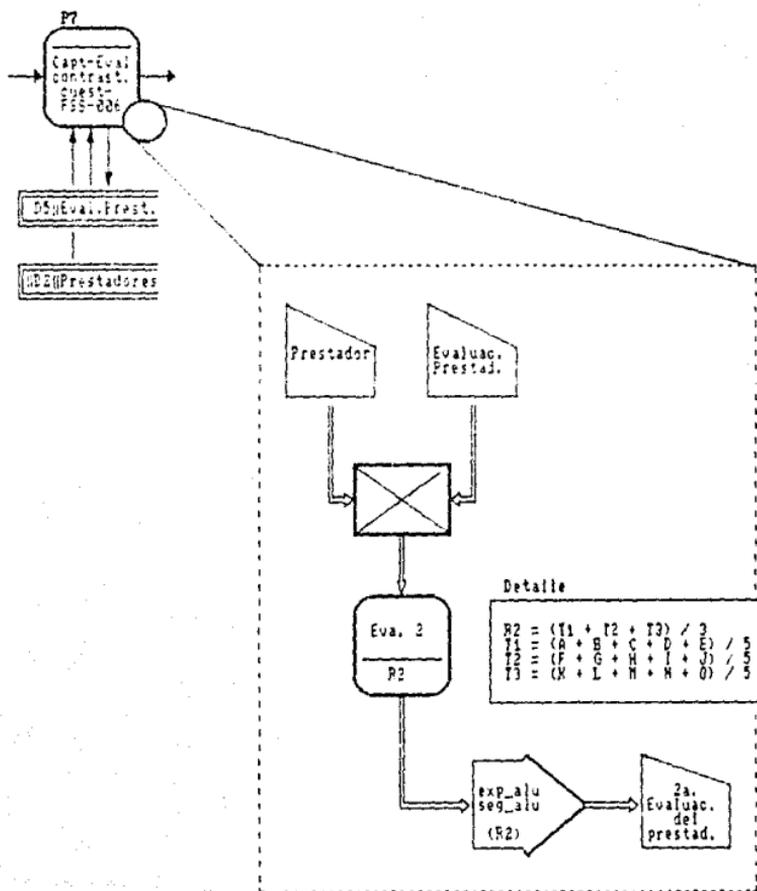


FIG. 4.7 Flujoograma 6: Segunda contrastación del prestador

El Flujoograma 7 de la Fig. 4.8 hace referencia al proceso de Captura de la Tercera Contratación del Prestador, o contrastación Cuestionarios - Programa de servicio social (proceso P8, fig. 2.9).

Para comenzar se realiza una junta natural de los archivos Prestador y Evaluación del Prestador, cuyo resultado se vuelve a unir con el archivo de Programas.

Posteriormente se lleva a cabo la Captura de la Tercera Contratación (variables A, B, C, D y E los cuales corresponden a valores que van del 0 al 10 utilizados para evaluar la correspondencia de la contratación obtenida en el flujoograma anterior, y los objetivos planteados por el programa de servicio social). Con esto se genera una Evaluación Final del Prestador, realizando las operaciones que se encuentran detalladas.

Para terminar se hace una proyección de los resultados obtenidos almacenándose en el archivo Evaluación del Prestador.

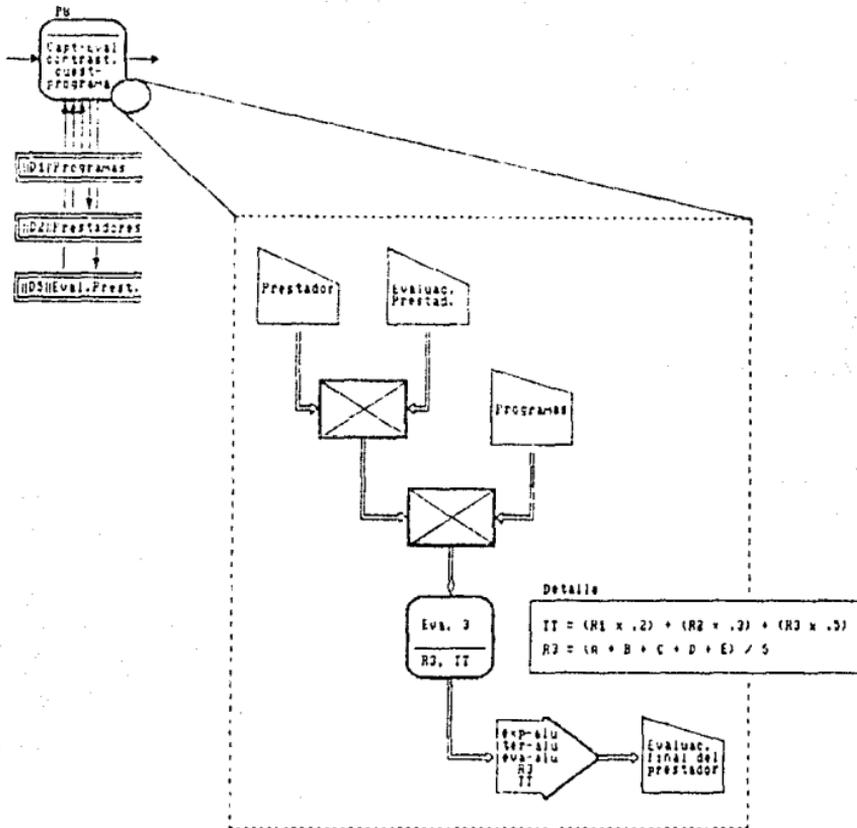


FIG. 4.9 Flujoograma 7: Tercera contrastación del prestador

La Fig. 4.9 muestra el Flujograma 8 correspondiente al proceso de Emisión de Cartas de Terminación del servicio social (proceso P9, fig. 2.9).

Este flujograma comienza con una serie de juntas naturales, primera entre los archivos de Programas y de Tablas; luego, del resultado obtenido con el archivo Prestador, y por último, de este mismo resultado con el archivo de Evaluación del Prestador.

Una vez realizadas estas juntas, se analiza el resultado final de la evaluación del prestador (TT). Si dicho valor es mayor o igual a 6, el servicio social se da por concluido, obteniendo su respectiva Carta de Terminación. En caso contrario, se dice que el servicio social no ha sido acreditado, quedando así el prestador imposibilitado de obtener su carta de terminación.

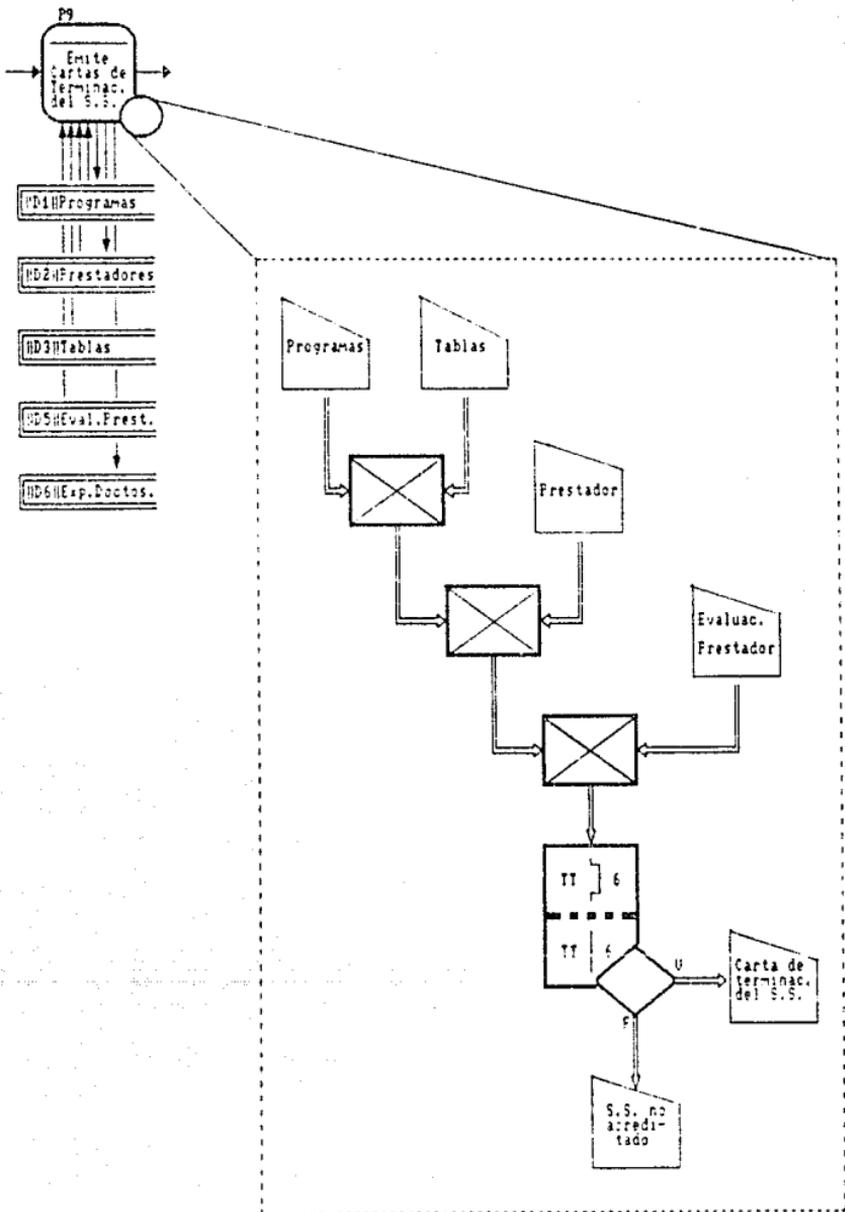


FIG. 4.9 Flujograma B: Generación de cartas de terminación

El Flujograma 9 de la Fig. 4.10, corresponde a las Consultas y Reportes eventuales que permitirá generar el sistema (proceso P10, fig. 2.9).

Este flujograma comienza con una serie de juntas naturales sucesivas entre todos los archivos generados por el sistema (excepto el de Expediente de Documentos), y termina con otra serie sucesiva de proyecciones que presentan en pantalla o en forma impresa una lista de los atributos requeridos.

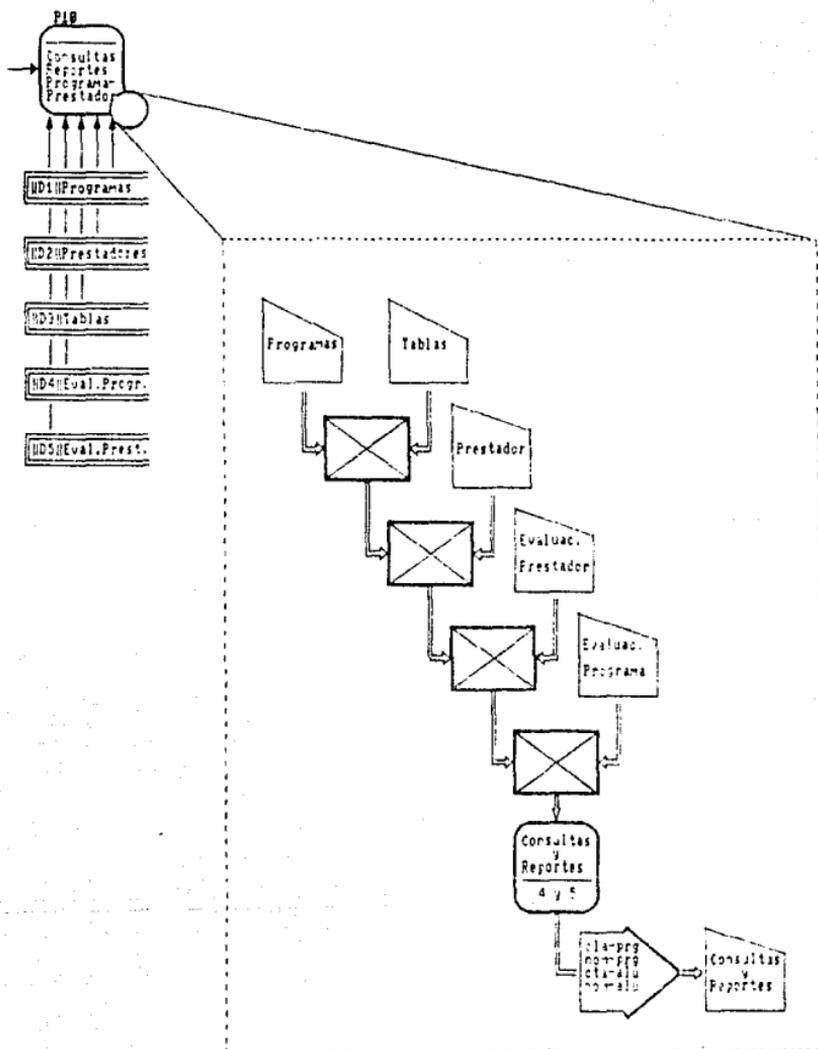


FIG. 4.18 Flujoograma 9: Generación de consultas y reportes

## CONCLUSIONES

El presente trabajo muestra la aplicación de las técnicas de Ingeniería de Software en el análisis y diseño de un sistema de información. Asimismo, la base de datos se encuentra en 3FN. Como se observa, entre más grande y complicado sea el sistema a diseñar, más útiles serán estas herramientas. El resultado final de este trabajo muestra el desarrollo de los principales programas del sistema en el Lenguaje de Flujograma de S. Chapa.

La principal contribución del trabajo, es su desarrollo en un nivel alto de abstracción, lo cual permitirá producir código en dBase.

Además, se presenta una nueva metodología (S. Chapa) para el desarrollo de sistemas en donde sólo es necesario plantear los problemas en términos gráficos sin necesidad de desarrollar los programas en el sentido tradicional de líneas de código. Esto muestra facilidades de representación desde el momento de establecer los requerimientos con el modelo de procesos hasta la generación automática de código. De esta forma, se tiene un punto de vista unificado del desarrollo del sistema utilizando exclusivamente descripciones gráficas en modelos de flujo de datos.

**BIBLIOGRAFIA**

1. Byers, Robert A.; "Everyman's Database Primer Featuring dBase III Plus"; Management Information Source Inc.; 1986.
2. Chapa V., Sergio; "Lenguaje de Flujograma para la Consulta en Base de Datos (Un Estudio Comparativo)."; Memorias de la Quinta Conferencia Internacional Las Computadoras en las Instituciones de Educación y de Investigación; UNISYS-UNAM; 1989.
3. Chapa V., Sergio; "Programación Algorítmica a Partir de Descriptores de Flujo de Información"; Tesis Doctoral, CINVESTAV; (por publicar).
4. Codd, E.F.; "A Relational Model for Large Shared Databanks"; ACM 13, No. 6, Association for Computing Machinery, 1970.
5. Date, C. J.; "Introducción a los Sistemas de Bases de Datos"; Addison-Wesley Iberoamericana; 1986.
6. Gane, Chris; Sarson, Trish; "Análisis Estructurado de Sistemas"; El Ateneo; 1987.
7. Jones, Edward.; "Using dBase III Plus"; McGraw-Hill; 1987.
8. Martin, James; "Organización de las Bases de Datos"; Prentice-Hall Hispanoamericana; 1977.
9. Senn, James A.; "Análisis y Diseño de Sistemas de Información"; Mc Graw Hill, 1987.

## A P E N D I C E

```

*
*   PROGRAMA: CAPPRG.PRG
*   FUNCION:  ACTUALIZACION DE PROGRAMAS
*
SET PROCEDURE TO CAPPRG

SELECT 2
USE CAT_PRG1 INDEX CAT_PRG1
REINDEX

DO WHILE .T.
  CLEAR
  DO PANPRG
  @ 3,1 SAY "Actualizacion de Programas"
  @ 5,33 SAY "NOMBRE DEL PROGRAMA"
  WOPC=" "

  STORE 0 TO WPRG, WTIPO, WCANAL, WCARAC, WAREA, WSUBAR, WACTIV
  STORE 0 TO WAPOYO, WSEFM, WSTAT, WFLSOL, WPLDIS
  STORE 0 TO WCAR1, WCAR2, WCAR3, WCAR4, WCAR5
  STORE 0 TO WCAR6, WCAR7, WCAR8, WCAR9, WCAR0
  STORE SPACE(40) TO WNPRG, WNDEP
  STORE SPACE(25) TO WUNRES, WREDEP, WDODEP, WTEDEP
  STORE SPACE(25) TO WREPRG, WPTPRG, WUBFRE, WTEPRE
  STORE SPACE(25) TO WOBPRG, WHOPRG, WRQPRG, WOSPRG
  STORE CTOD(" / / ") TO WFEINI, WFEFIN, WFEFUL

  IF WOPC <> "X"
    @ 9,1 CLEAR TO 20,78
  ENDIF

  @ 22,1 CLEAR TO 23,78
  @ 7,15 SAY SPACE(63)
  @ 7,7 GET WOPC PICTURE "!"
  READ

  IF WOPC <> "A" .AND. WOPC <> "B" .AND. WOPC <> "C";
    .AND. WOPC <> "X" .AND. WOPC <> "F"
    @ 22,1 SAY "Movimiento invalido. Debe ser A, B, C, X o F"
    WAIT CHR(186) + "Oprima cualquier tecla para continuar..."
    LOOP
  ENDIF

  IF WOPC = "A" .OR. WOPC = "B" .OR. WOPC = "C"
    @ 7,19 GET WPRG PICTURE "999,999" RANGE 1,999999
    READ
    IF WPRG < 0 .OR. WPRG = 0
      @ 22,1 SAY "Numero invalido. Debe ser mayor que cero"
      WAIT CHR(186) + "Oprima cualquier tecla para continuar..."
    LOOP
  ENDIF

```

```

ENDIF
ENDIF

```

```

DO CASE

```

```

CASE WOPC = "F"
  CLOSE ALL
  RETURN

```

```

CASE WOPC = "A"
  SELECT 2
  USE CAT_PRG1 INDEX CAT_PRG1
  REINDEX
  SEEK WPRG

```

```

  IF FOUND()
    @ 22,1 SAY "Numero de programa ya existe"
    WAIT CHR(186) + "Oprima cualquier tecla para continuar..."
    LOOP
  ENDIF

```

```

DO VALIDA

```

```

CASE WOPC = "B"
  SELECT 2
  USE CAT_PRG1 INDEX CAT_PRG1
  REINDEX
  SEEK WPRG

```

```

  IF .NOT. FOUND()
    @ 22,1 SAY "Numero de programa no existe"
    WAIT CHR(186) + "Oprima cualquier tecla para continuar..."
    LOOP
  ENDIF

```

```

DO PRESENTA

```

```

WPREG = "S"
@ 23,1 SAY "Desea dar la baja?... (S/N) " GET WPREG PICTURE
READ
@ 7,15 SAY SPACE(63)
@ 22,1 CLEAR TO 23,78

```

```

IF WPREG = "S"
  SELECT 2
  USE CAT_PRG1 INDEX CAT_PRG1
  REINDEX
  SEEK WPRG
  DELETE
  PACK
  SELECT 3
  USE CAT_PRG2 INDEX CAT_PRG2
  SEEK WPRG
  DELETE
  PACK
ENDIF

```

```

CASE WOPC = "C"

```

```

SELECT 2
USE CAT_PRG1 INDEX CAT_PRG1
REINDEX
SEEK WPRG

```

```

IF .NOT. FOUND()
    @ 22,1 SAY "Numero de programa no existe"
    WAIT CHR(186) + "Oprima cualquier tecla para continuar..."
    LOOP
ENDIF

```

```

WNPRG = NOM_PRG
WNDEP = DEP_PRG
WTIPO = TIPO
WCANAL = CANAL
WCARAC = CARACTER
WAREA = AREA
WSUBAR = SUBAREA
WACTIV = ACTIVIDAD
WAPOYO = APOYO
WCAR1 = CARRERA1
WCAR2 = CARRERA2
WCAR3 = CARRERA3
WCAR4 = CARRERA4
WCAR5 = CARRERA5
WCAR6 = CARRERA6
WCAR7 = CARRERA7
WCAR8 = CARRERA8
WCAR9 = CARRERA9
WCAR0 = CARRERA0
WSEM = SEMESTRE
WSTAT = STA_PRG

```

```

SELECT 3
USE CAT_PRG2 INDEX CAT_PRG2
REINDEX
SEEK WPRG

```

```

WUNRES = UNI_RES
WREDEP = RES_DEP
WDODEP = DOM_DEP
WTEDEP = TEL_DEP
WREPRG = RES_PRG
WPTRG = PTO_PRG
WUBPRE = UBI_PRE
WTEPRE = TEL_PRE
WFEINI = FEC_INI
WFEFIN = FEC_FIN
WOBPRG = OBJ_PRG
WPLSOL = PLA_SOL
WPLOIS = PLA_DIS
WHOPRG = HOR_PRG
WRQPRG = REQ_PRG
WOSPRG = OBS_PRG

```

```
DO VALIDA
```

```
CASE WOPC = "X"
```

```

SELECT 2
USE CAT_PRG1 INDEX CAT_PRG1
REINDEX--
SEEK WPRG

IF .NOT. FOUND()
  GO TOP
ENDIF

WRESP = "S"
WREN = 9

DO WHILE .NOT. EOF()
  IF WREN < 21
    @ WREN,19 SAY CLA_PRG PICTURE "999,999"
    @ WREN,33 SAY NOM_PRG PICTURE "!!!!!!!!!!!!!!!!!!!!!!";
    !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"
    WREN = WREN + 1
    SKIP
    ELSE
      @ 23,1 SAY "Desea continuar?... (S/N) ";
      GET WRESP PICTURE "!"
      READ
      @ 22,1 CLEAR TO 23,78
      IF WRESP <> "S"
        EXIT
        ELSE
          @ 9,1 CLEAR TO 20,78
          WREN = 9
          ENDIF
      ENDIF
    ENDDO

    WRESP = " "
    @ 23,1 SAY "D = Consulta a detalle      F = Fin"
    @ 23,50 GET WRESP PICTURE "!"
    READ
    IF WRESP = "D"
      @ 9,1 CLEAR TO 20,78
      @ 7,19 GET WPRG PICTURE "999,999" RANGE 1,999999
      READ
      @ 22,1 CLEAR TO 23,78
      IF WPRG <> 0
        SEEK WPRG
        IF .NOT. FOUND()
          @ 22,1 SAY "Numero de programa no existe"
          WAIT CHR(186) + "Oprima cualquier tecla ;
para continuar..."
          ELSE
            DO PRESENTA
            ENDIF
          ENDIF
        ENDIF
      @ 22,1 CLEAR TO 23,78

    ENDCASE
  ENDDO

```

ENDDO  
RETURN

\*\*\*\*\*  
\* RUTINA VALIDA  
\*\*\*\*\*

PROCEDURE VALIDA

```

WVAL = "N"
DO WHILE WVAL = "N"
  @ 22,1 CLEAR TO 23,78
  @ 7,33 GET WNPRG PICTURE "!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"
  @ 9,2 SAY "Dependencia: " GET WNDEP " GET WUNDEP
  @ 10,2 SAY "Unidad Responsable: " GET WUNRES " GET WUNRES
  @ 11,2 SAY "Responsable en la Dependencia:" GET WREDEF " GET WREDEF
  @ 12,2 SAY "Domicilio de la Dependencia: " GET WDDOEP " GET WDDOEP
  @ 13,2 SAY "Telefono en la Dependencia: " GET WTEDEP " GET WTEDEP
  @ 14,2 SAY "Responsable del Programa: " GET WREPRG " GET WREPRG
  @ 15,2 SAY "Puesto del Responsable: " GET WPTPRG " GET WPTPRG
  @ 16,2 SAY "Ubicacion del Prestador: " GET WUBPRE " GET WUBPRE
  @ 17,2 SAY "Telefono del Prestador: " GET WTEPRE " GET WTEPRE
  @ 18,2 SAY "Duracion del Programa. Del:" GET WFEINI " GET WFEINI
  @ 19,46 SAY "Al:" GET WFEFIN
  @ 19,2 SAY "Objetivo del Programa: " GET WOBPRG " GET WOBPRG
  @ 20,2 SAY "Horario: " GET WHOPRG " GET WHOPRG
READ
  IF READKEY() = 12
    EXIT
  ENDIF
  IF WNPRG = SPACE(40)
    @ 22,1 SAY "Nombre del programa en blanco "
    WAIT CHR(186) + "Oprima cualquier tecla para continuar..."
    LOOP
  ENDIF
  IF WNDEP = SPACE(40)
    @ 22,1 SAY "Nombre de la dependencia en blanco "
    WAIT CHR(186) + "Oprima cualquier tecla para continuar..."
    LOOP
  ENDIF
  WVAL = "S"
ENDDO
IF READKEY() <> 12
  @ 22,1 SAY " "
  WAIT CHR(186) + "Oprima cualquier tecla para continuar..."
  @ 9,1 CLEAR TO 20,78
  WVAL = "N"
DO WHILE WVAL = "N"
  @ 22,1 CLEAR TO 23,78
  @ 9,2 SAY "Requisitos: " GET WRQPRG " GET WRQPRG
  @ 10,2 SAY "Observaciones: " GET WOSPRG " GET WOSPRG
  @ 11,2 SAY "Plazas Sol.: " GET WPLSOL ; " GET WPLSOL ;

```

```

PICTURE "99" RANGE 1,99
      0 11,33 SAY "Plazas Disp.: " GET WPLDIS ;
PICTURE "99" RANGE 0,99
      0 12,2 SAY "Tipo de Programa: " GET WTIPO ;
PICTURE "99" RANGE 1,99
      0 13,2 SAY "Canal de Funcionamiento:" GET WCANAL ;
PICTURE "99" RANGE 1,99
      0 14,2 SAY "Caracter: " GET WCARAC ;
PICTURE "99" RANGE 1,99
      0 15,2 SAY "Area de Ubicacion: " GET WAREA ;
PICTURE "99" RANGE 1,99
      0 16,2 SAY "Subarea de Ubicacion: " GET WSUBAR ;
PICTURE "99" RANGE 1,99
      0 17,2 SAY "Actividades: " GET WACTIV ;
PICTURE "99" RANGE 1,99
      0 18,2 SAY "Apoyo del Programa: " GET WAPOYO ;
PICTURE "99" RANGE 1,99
      0 19,2 SAY "Semestre Solicitado: " GET WSEM ;
PICTURE "99" RANGE 1,99
      READ

      IF READKEY() = 12
          EXIT
      ENDIF

      SELECT 1
      USE TAB_TIPO INDEX TAB_TIPO
      REINDEX
      SEEK WTIPO
      IF .NOT. FOUND()
          0 22,1 SAY "Tipo de programa inexistente"
          WAIT CHR(186) + "Oprima cualquier tecla para continuar..."
          LOOP
      ELSE
          0 12,33 SAY NOM TIPO PICTURE
          "!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"
          ENDIF

      SELECT 1
      USE TAB_CANA INDEX TAB_CANA
      REINDEX
      SEEK WCANAL
      IF .NOT. FOUND()
          0 22,1 SAY "Canal de funcionamiento del programa inexistente"
          WAIT CHR(186) + "Oprima cualquier tecla para continuar..."
          LOOP
      ELSE
          0 13,33 SAY NOM CANAL PICTURE
          "!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"
          ENDIF

      SELECT 1
      USE TAB_CARA INDEX TAB_CARA
      REINDEX
      SEEK WCARAC
      IF .NOT. FOUND()
          0 22,1 SAY "Caracter del programa inexistente"
          WAIT CHR(186) + "Oprima cualquier tecla para continuar..."

```



```

SELECT 1
USE TAB_SEM INDEX TAB_SEM
REINDEX
SEEK WSEM
IF .NOT. FOUND()
    @ 22,1 SAY "Semestre inexistente"
    WAIT CHR(186) + "Oprima cualquier tecla para continuar..."
    LOOP
ELSE
    @ 19,33 SAY NOM_SEM PICTURE "!!!!!!!!!!!!"
ENDIF
WVAL = "S"
ENDDO
ENDIF

IF READKEY() <> 12
    @ 22,1 SAY " "
    WAIT CHR(186) + "Oprima cualquier tecla para continuar..."
    @ 9,1 CLEAR TO 20,78
    WVAL = "N"

DO WHILE WVAL = "N"
    @ 22,1 CLEAR TO 23,78
    @ 9,2 SAY "Carrera 1: " GET WCAR1 PICTURE "99" RANGE 1,99
    @ 10,2 SAY "Carrera 2: " GET WCAR2 PICTURE "99" RANGE 0,99
    @ 11,2 SAY "Carrera 3: " GET WCAR3 PICTURE "99" RANGE 0,99
    @ 12,2 SAY "Carrera 4: " GET WCAR4 PICTURE "99" RANGE 0,99
    @ 13,2 SAY "Carrera 5: " GET WCAR5 PICTURE "99" RANGE 0,99
    @ 14,2 SAY "Carrera 6: " GET WCAR6 PICTURE "99" RANGE 0,99
    @ 15,2 SAY "Carrera 7: " GET WCAR7 PICTURE "99" RANGE 0,99
    @ 16,2 SAY "Carrera 8: " GET WCAR8 PICTURE "99" RANGE 0,99
    @ 17,2 SAY "Carrera 9: " GET WCAR9 PICTURE "99" RANGE 0,99
    @ 18,2 SAY "Carrera 10: " GET WCAR0 PICTURE "99" RANGE 0,99
    READ

    IF READKEY() = 12
        EXIT
    ENDIF

    SELECT 1
    USE TAB_CAR INDEX TAB_CAR
    REINDEX
    SEEK WCAP1
    IF .NOT. FOUND()
        @ 22,1 SAY "Carrera 1 inexistente"
        WAIT CHR(186) + "Oprima cualquier tecla para continuar..."
        LOOP
    ELSE
        @ 9,33 SAY NOM_CAR PICTURE
        "!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"
        ENDIF

        IF WCAR2 > 0
            GO TOP
            SELK WCAR2
            IF .NOT. FOUND()
                @ 22,1 SAY "Carrera 2 inexistente"
                WAIT CHR(186) + "Oprima cualquier tecla para continua

```





ENDIF 9,1 CLEAR TO 20,78

ENDDO

ENDIF

SELECT 2

USE CAT\_PRG1 INDEX CAT\_PRG1

REINDEX

IF READKEY() <> 12

IF WOPC = "A"

APPEND BLANK

REPLACE CLA\_PRG WITH WPRG

ELSE

SEEK WPRG

ENDIF

REPLACE CLA\_PRG WITH WPRG

REPLACE NOM\_PRG WITH WNPRG

REPLACE DEP\_PRG WITH WNDEP

REPLACE TIPO WITH WTIPO

REPLACE CANAL WITH WCANAL

REPLACE CARACTER WITH WCARAC

REPLACE AREA WITH WAREA

REPLACE SUBAREA WITH WSUBAR

REPLACE ACTIVIDAD WITH WACTIV

REPLACE APOYO WITH WAPOYO

REPLACE SEMESTRE WITH WSEM

REPLACE CARRERA1 WITH WCAR1

REPLACE CARRERA2 WITH WCAR2

REPLACE CARRERA3 WITH WCAR3

REPLACE CARRERA4 WITH WCAR4

REPLACE CARRERA5 WITH WCAR5

REPLACE CARRERA6 WITH WCAR6

REPLACE CARRERA7 WITH WCAR7

REPLACE CARRERA8 WITH WCAR8

REPLACE CARRERA9 WITH WCAR9

REPLACE CARRERA0 WITH WCAR0

ENDIF

SELECT 3

\*IF READKEY() <> 12

USE CAT\_PRG2 INDEX CAT\_PRG2

REINDEX

IF WOPC = "A"

APPEND BLANK

REPLACE CLA\_PRG WITH WPRG

ELSE

SEEK WPRG

ENDIF

REPLACE CLA\_PRG WITH WPRG

REPLACE UNI\_RES WITH WUNRES

REPLACE RES\_DEP WITH WREDEP

REPLACE DOM\_DEP WITH WDODEP

REPLACE TEL\_DEP WITH WTEDEP

REPLACE RES\_PRG WITH WREPRG

REPLACE PTO\_PRG WITH WPTPRG

REPLACE UBI\_PRE WITH WUBPRE

REPLACE TEL\_PRE WITH WTEPRE

REPLACE FEC\_INI WITH WFEINI



```

@ 17,2 SAY "Actividades:                " + LTRIM(STR(ACTIVIDAD))
@ 18,2 SAY "Apoyo del Programa:         " + LTRIM(STR(APOYO))
@ 19,2 SAY "Semestre Solicitado:        " + LTRIM(STR(SEMESTRE))

```

```

WTIPO=TIPO
WCANAL = CANAL
WCARAC = CARACTER
WAREA = AREA
WSUBAR = SUBAREA
WACTIV = ACTIVIDAD
WAPOYO = APOYO
WCAR1 = CARRERA1
WCAR2 = CARRERA2
WCAR3 = CARRERA3
WCAR4 = CARRERA4
WCAR5 = CARRERA5
WCAR6 = CARRERA6
WCAR7 = CARRERA7
WCAR8 = CARRERA8
WCAR9 = CARRERA9
WCARO = CARRERA0
WSEM = SEMESTRE

```

```

SELECT 1
USE TAB_TIPO INDEX TAB_TIPO
REINDEX
SEEK WTIPO
@ 12,40 SAY NOM_TIPO PICTURE "!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"

USE TAB_CANAL INDEX TAB_CANAL
REINDEX
SEEK WCANAL
@ 13,40 SAY NOM_CANAL PICTURE "!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"

USE TAB_CARAC INDEX TAB_CARAC
REINDEX
SEEK WCARAC
@ 14,40 SAY NOM_CARAC PICTURE "!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"

USE TAB_AREA INDEX TAB_AREA
REINDEX
SEEK WAREA
@ 15,40 SAY NOM_AREA PICTURE "!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"

USE TAB_SUBA INDEX TAB_SUBA
REINDEX
SEEK WSUBAR
@ 16,40 SAY NOM_SUBAR PICTURE "!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"

USE TAB_ACTI INDEX TAB_ACTI
REINDEX
SEEK WACTIV
@ 17,40 SAY NOM_ACTIV PICTURE "!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"

USE TAB_APOY INDEX TAB_APOY
REINDEX
SEEK WAPOYO
@ 18,40 SAY NOM_APOYO PICTURE "!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"

```

```

USE TAB_SEM INDEX TAB_SEM
REINDEX
SEEK WSEM
@ 19,40 SAY NOM_SEM PICTURE "!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"

@ 22,1 SAY " "
WAIT CHR(186) + "Oprima cualquier tecla para continuar..."
@ 22,1 CLEAR TO 23,78
@ 9,1 CLEAR TO 20,78

SELECT 2
USE CAT_PRG1 INDEX CAT_PRG1
REINDEX
SEEK WPRG
@ 9,2 SAY "Carrera 1: " + LTRIM(STR(CARRERA1))
@ 10,2 SAY "Carrera 2: " + LTRIM(STR(CARRERA2))
@ 11,2 SAY "Carrera 3: " + LTRIM(STR(CARRERA3))
@ 12,2 SAY "Carrera 4: " + LTRIM(STR(CARRERA4))
@ 13,2 SAY "Carrera 5: " + LTRIM(STR(CARRERA5))
@ 14,2 SAY "Carrera 6: " + LTRIM(STR(CARRERA6))
@ 15,2 SAY "Carrera 7: " + LTRIM(STR(CARRERA7))
@ 16,2 SAY "Carrera 8: " + LTRIM(STR(CARRERA8))
@ 17,2 SAY "Carrera 9: " + LTRIM(STR(CARRERA9))
@ 18,2 SAY "Carrera 10: " + LTRIM(STR(CARRERA10))

SELECT 1
USE TAB_CAR INDEX TAB_CAR
REINDEX
SEEK WCAR1
@ 9,40 SAY NOM_CAR PICTURE "!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"

SEEK WCAR2
@ 10,40 SAY NOM_CAR PICTURE "!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"

SEEK WCAR3
@ 11,40 SAY NOM_CAR PICTURE "!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"

SEEK WCAR4
@ 12,40 SAY NOM_CAR PICTURE "!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"

SEEK WCAR5
@ 13,40 SAY NOM_CAR PICTURE "!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"

SEEK WCAR6
@ 14,40 SAY NOM_CAR PICTURE "!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"

SEEK WCAR7
@ 15,40 SAY NOM_CAR PICTURE "!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"

SEEK WCAR8
@ 16,40 SAY NOM_CAR PICTURE "!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"

SEEK WCAR9
@ 17,40 SAY NOM_CAR PICTURE "!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"

SEEK WCAR0
@ 18,40 SAY NOM_CAR PICTURE "!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"

```

```
IF WOPC= "X"  
  @ 22,33 SAY " "  
  WAIT CHR(166) + "Oprima cualquier tecla para continuar..."  
  @ 9,1 CLEAR TO 20,78  
ENDIF  
RETURN
```

```

*
*   PROGRAMA: CAPPRE.PRG
*   FUNCION:  ACTUALIZACION DE PRESTADORES
*

```

```
SET PROCEDURE TO CAPPRE
```

```
SELECT 2
USE CAT_PRES INDEX CAT_PRES
REINDEX
```

```
DO WHILE .T.
```

```

  CLEAR
  DO PANPRG
  @ 3,1 SAY "Actualizacion de Prestadores"
  @ 5,33 SAY "NOMBRE DEL PRESTADOR"
  WOPC=" "

```

```

  STORE 0 TO WCTA, WCAR, WSEM, WPRG, WEXP
  STORE SPACE(40) TO WHOM, WDOM, WNCAR
  STORE SPACE(25) TO WTEL

```

```

  IF WOPC <> "X"
    @ 9,1 CLEAR TO 20,78
  ENDIF

```

```

  @ 22,1 CLEAR TO 23,78
  @ 7,15 SAY SPACE(63)
  @ 7,7 GET WOPC PICTURE "!"
  READ

```

```

  IF WOPC <> "A" .AND. WOPC <> "B" .AND. WOPC <> "C";
    .AND. WOPC <> "X" .AND. WOPC <> "F"
    @ 22,1 SAY "Movimiento invalido. Debe ser A, B, C, X o F"
    WAIT CHR(186) + "Oprima cualquier tecla para continuar..."
    LOOP
  ENDIF

```

```

  IF WOPC = "A" .OR. WOPC = "B" .OR. WOPC = "C"
    @ 7,19 GET WCTA PICTURE "9999999-9" RANGE 1,99999999
    READ
    IF WCTA < 0 .OR. WCTA = 0
      @ 22,1 SAY "Cuenta invalida. Debe ser mayor que cero"
      WAIT CHR(186) + "Oprima cualquier tecla para continuar..."
    LOOP
  ENDIF
  ENDIF

```

```
DO CASE
```

```

CASE WOPC = "F"
  CLOSE ALL
  RETURN

```

```

CASE WOPC = "A"
  SELECT 2
  USE CAT_PRES INDEX CAT_PRES
  REINDEX

```

```

SEEK WCTA

IF FOUND()
  @ 22,1 SAY "Cuenta del alumno ya existe"
  WAIT CHR(186) + "Oprima cualquier tecla para continuar..."
  LOOP
ENDIF

DO VALIDA

CASE WOPC = "B"
  SELECT 2
  USE CAT_PRES INDEX CAT_PRES
  REINDEX_
  SEEK WCTA

  IF .NOT. FOUND()
    @ 22,1 SAY "Cuenta del alumno no existe"
    WAIT CHR(186) + "Oprima cualquier tecla para continuar..."
    LOOP
  ENDIF

  DO PRESENTA

  WPREG = "S"
  @ 23,1 SAY "Desea dar la baja?... (S/N) " GET WPREG PICTURE
  READ
  @ 7,15 SAY SPACE(63)
  @ 22,1 CLEAR TO 23,78

  IF WPREG = "S"
    SELECT 2
    USE CAT_PRES INDEX CAT_PRES
    REINDEX_
    SEEK WCTA
    DELETE
    PACK
  ENDIF

CASE WOPC = "C"
  SELECT 2
  USE CAT_PRES INDEX CAT_PRES
  REINDEX_
  SEEK WCTA

  IF .NOT. FOUND()
    @ 22,i SAY "Cuenta del alumno no existe"
    WAIT CHR(186) + "Oprima cualquier tecla para continuar..."
    LOOP
  ENDIF

  WCTA = CTA_ALU
  WHOM = NOM_ALU
  WCAR = CAR_ALU
  WSEM = SEM_ALU
  WDOM = DOM_ALU
  WTEL = TEL_ALU
  WPRG = CLA_PRG

```

```

WEXP = EXP_ALU

SELECT 1
USE TAB_CAR INDEX TAB_CAR
REINDEX
SEEK WCAR

WHCAR = NOM_CAR

SELECT 3
USE CAT_PRG1 INDEX CAT_PRG1
REINDEX
SEEK WPRG

WHPRG = NOM_PRG

DO VALIDA

CASE WOPC = "X"
SELECT 2
USE CAT_PRES INDEX CAT_PRES
REINDEX
SEEK WCTA

IF .NOT. FOUND()
    GO TOP
ENDIF

WRESP = "S"
WREN = 9

DO WHILE .NOT. EOF()
    IF WREN < 21
        @ WREN,19 SAY CTA_ALU PICTURE "9999999-9"
        @ WREN,33 SAY NOM_ALU PICTURE "!!!!!!!!!!!!!!!!!!!!!!";
!!!!!!!!!!!!!!!!!!!!!!"
        WREN = WREN + 1
        SKIP
        ELSE
            @ 23,1 SAY "Desea continuar?... (S/N) ";
            GET WRESP PICTURE "!"
            READ
            @ 22,1 CLEAR TO 23,78
            IF WRESP <> "S"
                EXIT
            ELSE
                @ 9,1 CLEAR TO 20,78
                WREN = 9
            ENDIF
        ENDIF
    ENDDO

WRESP = " "
@ 23,1 SAY "D = Consulta a detalle      F = Fin"
@ 23,50 GET WRESP PICTURE "!"
READ
IF WRESP = "D"
    @ 9,1 CLEAR TO 20,78

```

```

@ 7,19 GET WCTA PICTURE "9999999-9" RANGE 1,9999999
READ
@ 22,1 CLEAR TO 23,78
IF WCTA <> 0
  SEEK WCTA
  IF .NOT. FOUND()
    @ 22,1 SAY "Numero de Cuenta no existe"
    WAIT CHR(186) + "Oprima cualquier tecla ;
para continuar..."
  ELSE
    DO PRESENTA
  ENDIF
ENDIF
ENDIF
@ 22,1 CLEAR TO 23,78
ENDCASE
ENDDO
RETURN

```

```

*****
* RUTINA VALIDA
*****

```

## PROCEDURE VALIDA

```

WVAL = "N"
DO WHILE WVAL = "N"
  @ 22,1 CLEAR TO 23,78
  @ 7,33 GET WNOM PICTURE "!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"
  @ 9,2 SAY "Carrera: " GET WCAR ;
  PICT "99" range 1,99
  @ 11,2 SAY "Semestre: " GET WSEM ;
  PICT "99" range 1,99
  @ 13,2 SAY "Domicilio: " GET WDOM
  @ 15,2 SAY "Telefono: " GET WTEL
  @ 17,2 SAY "Clave del Programa:" GET WPRG ;
  PICT "999999" range 1, 999999
  @ 19,2 SAY "No. de Expediente: " GET WEXP ;
  PICTURE "999999" rang 1,999999
  READ

  IF READKEY() = 12
    EXIT
  ENDIF

  IF WNOM = SPACE(40)
    @ 22,1 SAY "Nombre del Alumno en blanco "
    WAIT CHR(186) + "Oprima cualquier tecla para continuar..."
  LOOP
ENDIF

IF WEXP = 0
  @ 22,1 SAY "Numero de Expediente en blanco "
  WAIT CHR(186) + "Oprima cualquier tecla para continuar..."
  LOOP

```



```

REPLACE NOM_ALU WITH WDOM
REPLACE CAR_ALU WITH WCAR
REPLACE SEM_ALU WITH WSEM
REPLACE DOM_ALU WITH WDOM
REPLACE TEL_ALU WITH WTEL
REPLACE CLA_PRG WITH WPRG
REPLACE EXP_ALU WITH WEXP

```

```
ENDIF
```

```

@ 22,1 SAY " "
WAIT CHR(186) + "Oprima cualquier tecla para continuar..."

```

```
RETURN
```

```

*****
* RUTINA PRESENTA
*****

```

```
PROCEDURE PRESENTA
```

```

@ 7,19 SAY CTA_ALU PICTURE "9999999-9"
@ 7,33 SAY NOM_ALU PICTURE "!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"
@ 9,2 SAY "CaFrera: " + LTRIM(STR(CAR_ALU))
@ 11,2 SAY "Semestre: " + LTRIM(STR(SEM_ALU))
@ 13,2 SAY "Domicilio: " + DOM_ALU
@ 15,2 SAY "Telefono: " + TEL_ALU
@ 17,2 SAY "Clave del Programa: " + LTRIM(STR(CLA_PRG))
@ 19,2 SAY "No. de Expediente: " + LTRIM(STR(EXP_ALU))

```

```

WCAR = CAR_ALU
WSEM = SEM_ALU
WPRG = CLA_PRG

```

```

SELECT 1
USE TAB_CAR INDEX TAB_CAR
REINDEX
SEEK WCAR
@ 9,33 SAY NOM_CAR PICTURE "!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"

```

```

USE TAB_SEM INDEX TAB_SEM
REINDEX
SEEK WSEM
@ 11,33 SAY NOM_SEM PICTURE "!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"

```

```

SELECT 3
USE CAT_PRG1 INDEX CAT_PRG1
REINDEX
SEEK WPRG
@ 17,33 SAY NOM_PRG PICTURE "!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"
IF WOPC= "X"
@ 22,33 SAY " "
WAIT CHR(186) + "Oprima cualquier tecla para continuar..."
@ 17,1 CLEAR TO 20,78

```

```
ENDIF
```

```
RETURN
```

```
*  
* PROGRAMA: Fanprg.prg  
* FUNCION: Caratula de captura de programa y prestadores  
*
```

```
CLEAR  
@ 0,0 TO 24,79 DOUBLE  
@ 21,1 TO 21,78 DOUBLE  
@ 4,1 TO 4,78 DOUBLE  
@ 8,1 TO 8,78  
@ 1,1 SAY "Escuela Nacional de Estudios Profesionales Acatlan"  
@ 1,64 SAY "Fecha:"  
@ 1,71 SAY DATE()  
@ 2,1 SAY "Departamento de Servicio Social"  
@ 2,64 SAY "HORA: "  
@ 2,71 SAY TIME()  
@ 5,5 SAY "MOVTO:"  
@ 5,20 SAY "CLAVE"  
@ 6,2 SAY "(A,B,C,X,F)"  
@ 7,6 SAY "; ¿"
```

```
return
```