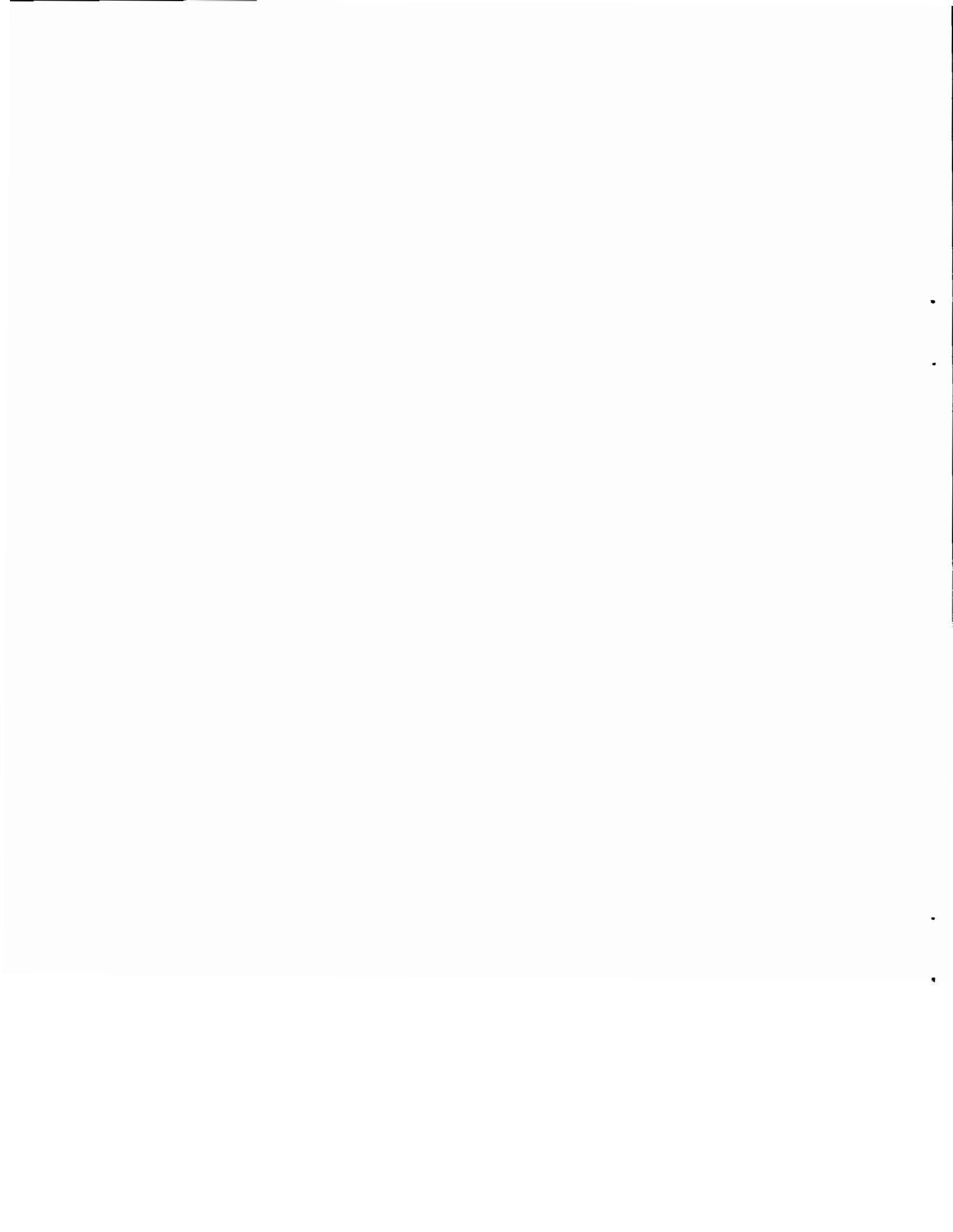


REDATAM

**MANUAL DE GENERACION
DE BASES
DE DATOS REDATAM**

NACIONES UNIDAS
COMISION ECONOMICA PARA AMERICA LATINA Y EL CARIBE
CENTRO LATINOAMERICANO DE DEMOGRAFIA



Distr.
GENERAL
LC/DEM/G.53
Serie A, No.178
Octubre de 1987
ORIGINAL: ESPAÑOL

REcuperación de DATos censales para Areas pequeñas por Microcomputador

(REDATAM)

Versión 2.00 (31 de marzo de 1987)

MANUAL DE GENERACION DE BASES DE DATOS REDATAM

Proyecto financiado por el Centro Internacional de Investigaciones para el Desarrollo (CIID) de Canadá, con el apoyo del Fondo de las Naciones Unidas para Actividades en Materia de Población (FNUAP) y la Agencia Canadiense para el Desarrollo Internacional (ACDI).

COMISION ECONOMICA PARA AMERICA LATINA Y EL CARIBE (CEPAL)

CENTRO LATINOAMERICANO DE DEMOGRAFIA (CELADE)

Casilla 91 - Santiago, Chile
Tel: 2283206 Cable: UNATIONS

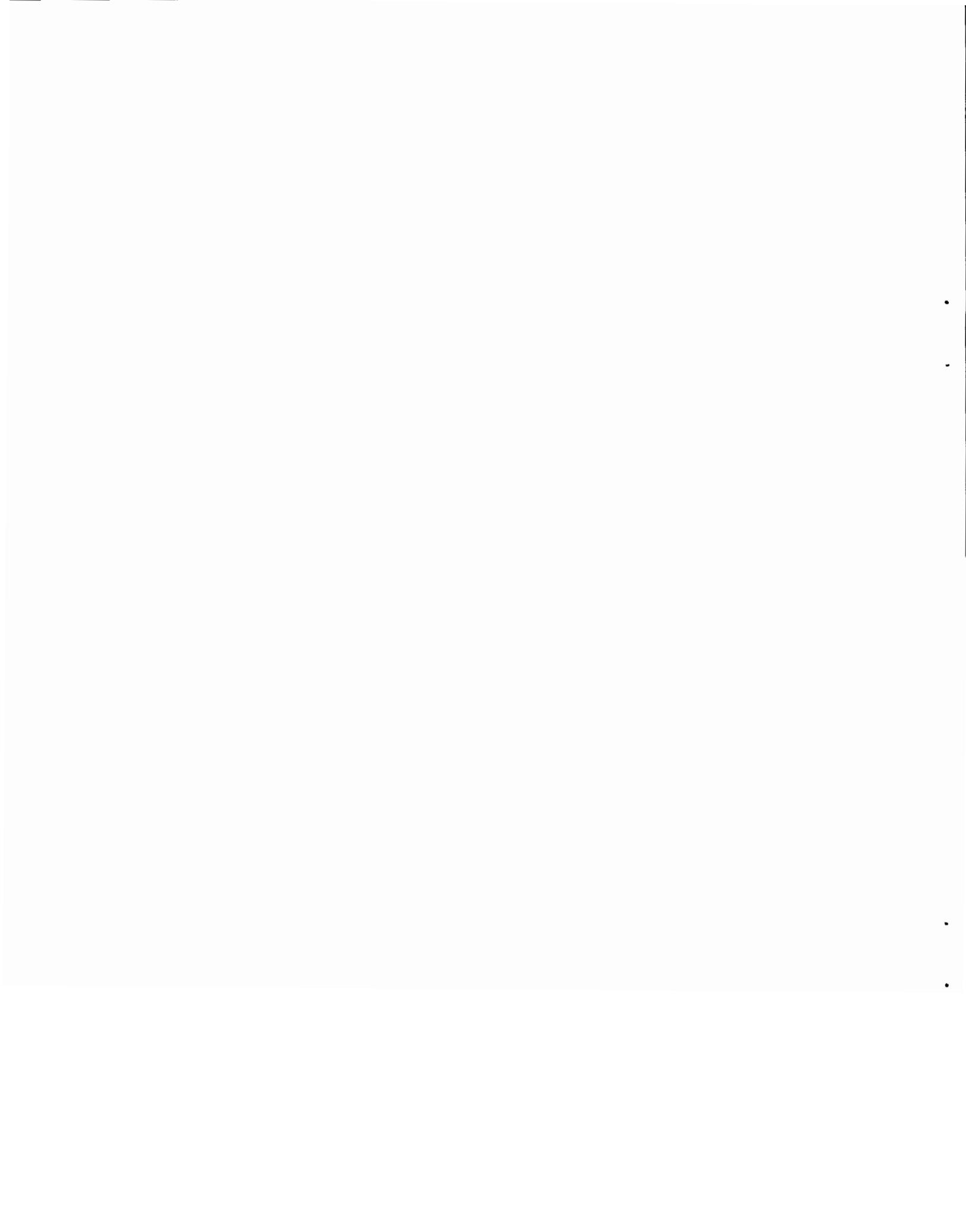


TABLA DE CONTENIDO

<u>INTRODUCCION</u>	1
1. <u>LOS DATOS INICIALES YA SE ENCUENTRAN EN EL MICROCOMPUTADOR</u>	2
1.1 Comentarios iniciales	2
1.2 Etapas específicas para GUYREDEM	2
1.3 Creación de la base de datos	3
1.3.1 Nombre de la base de datos	3
1.3.2 Ordenar el archivo de ingreso por orden geográfico	3
1.3.3 Creación del archivo .NOM	3
1.3.4 Frecuencias para el archivo de ingreso	4
1.3.5 Creación del Diccionario de Datos	4
1.3.6 Creación del archivo Geográfico	5
1.3.7 Generación de la Base de Datos.	5
2. <u>LOS DATOS ESTAN EN UN COMPUTADOR CENTRAL</u>	6
2.1 Requisitos para la generación de la base de datos	6
2.2 Comentarios iniciales	6
2.3 Etapas iniciales	7
2.4 Creación del Diccionario de Datos	10
2.5 Procesos a realizar en el computador central	10
2.6 Transmisión de los datos	11
2.7 Actualización del archivo de nombres	12
2.8 Generación del archivo geográfico	13
2.9 Generación de los archivos de índices	13
2.10 Generación de enlaces jerárquicos	14
2.11 Generación de archivos invertidos	15
2.12 Verificación del sistema	16
2.13 Respaldos	16
2.14 Requisitos finales de espacio	16
3. <u>AGREGANDO NUEVAS VARIABLES A UNA BASE DE DATOS YA EXISTENTE</u>	17
3.1 Situación 1: La variable puede derivarse de variables ya existentes	17
3.1.1 Procedimiento inicial	17
3.1.2 Creación de la variable	17
3.1.3 Actualización del diccionario	18
3.1.4 Generación de la base de datos	18
3.1.5 Comentarios	18
3.2 Situación 2: La variable es una nueva variable	18
APENDICE A.1 - PROGRAMA GUYCODE.BAS	19
APENDICE A.2 - PROGRAMA GUYNOM.BAS	21
APENDICE B - ESTIMACION DEL ESPACIO PARA LA BASE DE DATOS	22
APENDICE C - ARCHIVOS DEL SISTEMA	25
APENDICE D - EJEMPLO DE UNA HOJA DE CONTROL PARCIAL	26

REDATAM 2.00 Generación de Bases de Datos

APENDICE E	-	DIAGRAMA DEL TRABAJO EN EL 'MAINFRAME'	27
APENDICE F.1	-	PROGRAMA CORTES	28
APENDICE F.2	-	PROGRAMA VIVIENDA	33
APENDICE F.3	-	PROGRAMA POBLACION	44

INTRODUCCION

Este manual es un complemento del Manual del Usuario REDATAM.^{1/} Su objetivo principal es instruir al usuario, a través de ejemplos, en la creación de nuevas bases de datos.

El manual está dividido en tres partes, además de los apéndices.

La primera parte describe la creación de una base de datos de una pequeña encuesta, para lo cual los datos iniciales ya están grabados directamente en un medio accesible por microcomputador.

La segunda parte está dirigida a la creación de una base de datos a partir de un computador central, conectado al microcomputador. Esta parte debe ser usada para censos o encuestas cuyos archivos iniciales son demasiado grandes para ser almacenados en los medios normales de un microcomputador.

La tercera parte describe la forma de agregar nuevas variables a una base de datos ya existente.

^{1/} CELADE, Manual del Usuario REDATAM (versión 2.00). (LC/DEM/G.50), Santiago de Chile, 24 de junio de 1987.

1. LOS DATOS INICIALES YA SE ENCUENTRAN EN EL MICROCOMPUTADOR -----

1. LOS DATOS INICIALES YA SE ENCUENTRAN EN EL MICROCOMPUTADOR

1.1 Comentarios iniciales.

El ejemplo utilizado en esta sección fue una encuesta realizada en Guyana en 1986, llamada GUYREDEM, en la cual CELADE desempeñó un papel importante. Se escogió esta encuesta porque era un archivo relativamente pequeño que ya se encontraba en diskettes, lo cual significaba que todo el proceso podía efectuarse utilizando solamente microcomputadores.

La encuesta tiene aproximadamente 50.000 registros de igual tamaño, con alrededor de 42.000 personas y 8.000 viviendas. La encuesta completa está contenida en tres diskettes de alta densidad, con un archivo por cada Distrito de Enumeración (ED) en forma directa, sin "carriage return" (CR) y sin "line feed" (LF) al final de cada registro.

1.2 Etapas específicas para GUYREDEM.

1.2.1 El primer paso fue copiar los tres diskettes en el directorio llamado GUYREDEM.

1.2.2 Luego, se hizo un ordenamiento del directorio por nombre, ya que la estructura del nombre de archivo incluía la región, el código de "MAJOR AREA" y el número ED, para usarlos como variables geográficas en la base de datos REDATAM.

1.2.3 A continuación fue necesario crear un archivo "rectangular" conteniendo todos los registros, para lo cual se utilizó un programa Basic (GUYCODE.BAS), (ver Apéndice A.1). Este programa también se usó para insertar el peso del registro, calculado para cada región. Los archivos de salida fueron copiados en el mismo orden en que se encontraban, y este archivo de salida (GY86) debería estar ya ordenado por geografía (ver punto 1.2.2). Este archivo ocupa alrededor de 3.7 MB y tiene una estructura simple de dos tipos de registro, uno para vivienda y otro para personas. Tiene la siguiente estructura en lo que se refiere a los tipos de datos:

```
RECORD TYPE      ID variables...      DATA ...
```

1.2.4 El directorio completo de \GUYREDEM se respaldó en 7 diskettes de alta densidad.

1.2.5 Todos los archivos originales fueron borrados para tener espacio disponible para la base de datos y también los archivos temporales de SPSS que serían utilizados en una etapa posterior del procesamiento (ver punto 1.3.4). El SPSS.SY1 es muy grande, 4 bytes por cada variable tantas veces el número de registros.

1. LOS DATOS INICIALES YA SE ENCUENTRAN EN EL MICROCOMPUTADOR -----

1.3 Creación de la base de datos.

1.3.1 Nombre de la base de datos.

El nombre de la base de datos debe tener cuatro caracteres, comenzando con una letra. Se decidió usar "GY86".

1.3.2 Ordenar el archivo de ingreso por orden geográfico.

En el caso de GUYREDEM, esto no fue necesario porque el archivo de ingreso ya fue creado en forma ordenada.

Si este no fuera el caso, el archivo de ingreso debe ser ordenado utilizando como patrón el conjunto de variables de identificación de geografía, necesarias para el acceso geográfico a la base de datos de REDATAM.

1.3.3 Creación del archivo .NOM.

Este archivo debe tener todos los códigos geográficos y nombres que se encuentran en el archivo de ingreso. Con el fin de garantizar la precisión de los datos, la mejor forma es escribir un programa para leer el archivo de ingreso y producir el archivo .NOM con todos los códigos por niveles geográficos. Este programa también puede usarse para verificar la secuencia de ordenamiento del archivo de ingreso. En el caso de GUYREDEM, el programa se llama GUYNOM.BAS (ver Apéndice A.2.).

Este programa creó el archivo GY86.NOM, pero con dos pequeños problemas: el primero que no estaba en el orden requerido por el sistema REDATAM y el segundo que los nombres correspondientes a los niveles geográficos eran ficticios (como ED 234, REGION 01, etc.). Ambos fueron resueltos usando SideKick, programa que permitió ordenar los registros e ingresar los nombres apropiados, utilizando una lista entregada por el usuario. En ese momento, se descubrió que el archivo contenía algunos EDs que no estaban en la lista del usuario, probando de esta forma que la decisión de crear el archivo .NOM automáticamente desde el archivo de ingreso de datos fue una buena manera de detectar errores.

En realidad, habría sido más rápido si se hubiera incorporado esta función en el primer programa, creando el archivo de ingreso (GUYCODE.BAS).

El tercer paso fue verificar el tamaño máximo de los nombres para cada nivel de registro en el archivo .NOM, con el fin de informar más tarde al sistema REDATAM al crear el archivo GEO (ver punto 1.3.6 más abajo). Hay que tener cuidado al decidir el tamaño del nombre, porque REDATAM trunca los nombres al final si ellos no se ajustan a la pantalla de selección geográfica. El tamaño máximo del nombre varía para cada nivel, porque estos se muestran en la pantalla en diferentes columnas, dependiendo del tamaño de su código y de la posición en la estructura

1. LOS DATOS INICIALES YA SE ENCUENTRAN EN EL MICROCOMPUTADOR -----

geográfica. El primer nivel tiene más espacio para imprimir sus nombres que el segundo, porque a medida que se van agregando niveles cada uno de los que siguen van apareciendo indentados en la pantalla. Así es como el segundo nivel tendrá menos espacio que el primero y, a su vez, el tercer nivel tendrá menos espacio que el segundo.

En el caso de GUYREDEM, como tiene sólo tres niveles y mirando en el archivo GY86.NOM, decidimos utilizar 30 caracteres para los dos primeros niveles (Región y Área más Grande) y 33 caracteres para el tercer nivel (ED), siendo esta última cifra el máximo aceptado por REDATAM, sin que trunque los niveles que la poseen.

1.3.4 Frecuencias para el archivo de ingreso.

Es necesario tener una distribución de frecuencia para todas las variables del archivo de ingreso. Hay varias razones para esto:

1. Cuando se crea el diccionario de datos (ver punto 1.3.5 a continuación) es necesario conocer los rangos mínimos y máximos para cada variable.
2. Para esta etapa del diccionario de datos, también es necesario establecer los nombres por categorías de cada variable.
3. Si hay valores fuera de rango para las variables, es altamente deseable que ellos sean detectados con anterioridad. Si tales errores son muy frecuentes y estadísticamente significativos, el archivo debería ir a un proceso de "limpieza" antes de ser cargado en el formato REDATAM.
4. Estas frecuencias deberían compararse con las ejecutadas con el sistema REDATAM después de la generación de la base de datos, (ver punto 1.3.7) con el fin de verificar el proceso completo.

En el caso de GUYREDEM, se utilizó el sistema SPSS, con tres ejecuciones diferentes debido a las limitaciones del espacio en disco. Puede usarse cualquier otro paquete (e.g., SAS, SL-MICRO, e incluso un programa del usuario, aunque a la larga significa mucho más tiempo y no es aconsejable, excepto en el caso en que no haya ningún otro paquete disponible).

1.3.5 Creación del Diccionario de Datos.

Dentro del directorio de GUYREDEM y utilizando la opción Diccionario de Datos del menú principal del REDATAM, seleccionando luego la opción de Mantenimiento y finalmente la función de Creación, se requiere el nombre de la base de datos (i.e., GY86) y luego se comienza dando las entradas adecuadas.

Para cada variable hay que especificar el nombre, descripción, tipo de dato, largo original, posición inicial, valores mínimos y máximos y tipo de registro. El tipo de registro es 0 para las variables geográficas, 1 para las variables de vivienda y 2 para las variables de personas. El tipo de datos puede ser B (para binario, que significa que la base de datos será comprimida), o C para el tipo de carácter (los

1. LOS DATOS INICIALES YA SE ENCUENTRAN EN EL MICROCOMPUTADOR -----

datos no serán comprimidos en la etapa Autoload; ver punto 1.3.7). Los otros campos se explican por sí solos y pueden obtenerse del archivo de ingreso y de la distribución de frecuencias. El valor máximo del campo es usado por el sistema para calcular el tamaño de las variables comprimidas.

Para usar REDATAM es deseable tener un archivo .BAT con los parámetros apropiados, con el fin de tener acceso a la base de datos y a sus archivos. Utilizando SideKick (o cualquier otro editor), copiar el archivo MIRANDA.BAT (que está incluido en el REDATAM) y luego de acuerdo a esto cambiar el nombre y el directorio de la base de datos, según las necesidades de ésta.

1.3.6 Creación del archivo Geográfico.

Esta se hace utilizando la opción Mantención de la Base de Datos, extraída del menú principal del REDATAM y seleccionando luego la opción de Generación del archivo GEO. Pregunta por las variables geográficas y sus respectivos tamaños de nombre. Para GUYREDEM hubo tres variables, REGION con 30 caracteres, MAJAREA con 30 caracteres y ED con 33 caracteres. Este proceso no debería tomar más de un minuto o dos.

1.3.7 Generación de la Base de Datos.

Después de seleccionar la opción Mantención del menú principal del REDATAM y seleccionando luego el proceso Autoload, se debe dar el nombre de la base de datos (GY86) cuando es pedida. La base de datos es comprimida automáticamente (si el tipo de datos fue B en el punto 1.3.5).

Después de eso, es necesario obtener una distribución de frecuencias para cada variable para verificar que todas las variables fueron creadas en forma correcta y que los valores corresponden a las frecuencias hechas previamente utilizando el archivo de ingreso. Para obtener frecuencias usando REDATAM, es necesario crear un archivo de Selección Geográfica, seleccionando la encuesta completa (T's en el nivel más alto) y usar este archivo en el comando GEOGRAPHY del Procesador Estadístico, junto con el comando FREQUENCIES. Si una variable es contestada sólo por un cierto grupo de personas (por ejemplo, Educación se aplica a personas de 5 años y más), es necesario filtrar los correspondientes casos usando los apropiados comandos IF antes del comando FREQUENCIES.

El proceso Autoload tomó alrededor de 2 horas para un archivo completo (50,000 registros) en un IBM/AT. El proceso FREQUENCIES para las 39 variables llevó alrededor de 2 horas y 20 minutos.

La base de datos completa ocupa alrededor de 906 Kb de espacio de disco, 22 Kb para las variables de vivienda (4), 820 Kb para las variables de personas (32) y 64 Kb para los índices y control de archivos. Debe mencionarse que todas las variables están comprimidas en forma binaria.

2. LOS DATOS ESTAN EN UN COMPUTADOR CENTRAL -----

2. LOS DATOS ESTAN EN UN COMPUTADOR CENTRAL

2.1 Requisitos para la generación de la base de datos.

Para la instalación de una base de datos REDATAM de tamaño medio, por ejemplo, el Censo de Población y Vivienda de Costa Rica de 1984 con 2.5 millones de registros de personas, se requiere de cierta infraestructura mínima, la que se puede resumir en lo siguiente:

-un computador central capaz de procesar el archivo completo (ordenamientos y producción de marginales) y con la facilidad de poder conectarse a un microcomputador para el traspaso de información.

-un microcomputador, compatible con IBM, con suficiente espacio en disco (60 MB para el ejemplo de Costa Rica), una unidad de grabación de respaldos (cinta o diskette de 1.2 MB), una tarjeta (si es necesaria) para la comunicación y traspaso de archivos con el computador central. Este microcomputador debe estar disponible para esta instalación todo el tiempo, pues probablemente se utilice día y noche.

2.2 Comentarios iniciales.

Esta descripción resume las etapas principales en la generación de una base de datos REDATAM a partir de un censo o encuesta de tamaño intermedio, almacenado en un medio de grabación para "mainframe" (cinta magnética por ejemplo). Se usa como ejemplo la generación de la base de datos CR84, que contiene el Censo de Población y Vivienda de Costa Rica de 1984 con 500,000 registros de hogares y 2,500,000 de población, aproximadamente. Esta información estaba almacenada en registros de tamaño fijo de 80 bytes, ocupando 230 Mb, en seis cintas de 2,400 pies.

Los cálculos iniciales estimaron aproximadamente 51.5 Mb de espacio en disco (ver el Apéndice B para fórmula de cálculo) para almacenar esta base de datos, utilizando la posibilidad de REDATAM de procesar información en formato comprimido (almacenar para cada variable sólo los bits necesarios para su representación).

Antes de entrar al detalle de cada etapa a realizar, se puede resumir el trabajo a efectuar en los siguientes puntos:

- a) estudio de los datos a utilizar, obtención de frecuencias para la identificación de los códigos que se considerarán.
- b) en el computador central ("mainframe") se deben crear los siguientes archivos a partir de los datos del censo o encuesta:

-CORTES: archivo que contiene todos los cortes geográficos.

2. LOS DATOS ESTAN EN UN COMPUTADOR CENTRAL -----

-NOMBRES: archivo con los códigos de las áreas geográficas que llevarán nombre en REDATAM.

-PUNTEROS: archivo consistente en un registro por hogar, con el total de personas del hogar.

-ARCHIVOS DE DATOS DE VIVIENDA: se deben generar tantos archivos como variables de vivienda se incluyan en REDATAM, cada archivo contendrá solamente los valores de una variable determinada.

-ARCHIVOS DE DATOS DE POBLACION: se deben crear tantos archivos como variables de población se incluyan en REDATAM, cada archivo contendrá solamente los valores de una variable determinada.

- c) transferencia de los archivos anteriores al microcomputador.
- d) generación del diccionario de datos en REDATAM.
- e) agregar al archivo de NOMBRES los nombres de cada área geográfica.
- f) generación del archivo geográfico en REDATAM.
- g) generación de los archivos de índices en REDATAM.
- h) generación del archivo de punteros en REDATAM.
- i) generación de los archivos invertidos tanto para vivienda como para población.
- j) prueba del sistema.

En la generación de la base de datos REDATAM para el Censo de Costa Rica de 1984, se tardó aproximadamente un mes. Este tiempo depende primordialmente de la cantidad de registros, pues el proceso de transmisión de la información entre el "mainframe" y el microcomputador consume mucho tiempo.

2.3 Etapas iniciales.

El primer paso es la revisión de la información inicial. Hay que identificar exactamente la cantidad de registros a incluir, el formato de cada tipo de registro, los códigos utilizados, etc. Para esto es necesario obtener frecuencias de todas las variables a incluir, lo que permite detectar posibles valores fuera de rango.

Para el censo de Costa Rica se utilizó el programa SISTMARG, desarrollado por el CELADE para la obtención de las distribuciones de frecuencias de las variables del censo. Para las variables de vivienda se demoró 18 minutos de CPU y para las de población 46 minutos, en un computador IBM 3031.

2. LOS DATOS ESTAN EN UN COMPUTADOR CENTRAL -----

Una vez que se tienen todos los posibles valores de cada variable con exactitud, se debe definir lo siguiente:

- a) el valor mínimo y máximo y qué hacer con los valores fuera de estos rangos (blancos, ignorados y posibles inconsistencias). Con esta información ya se puede calcular la cantidad de bits necesarios para almacenar cada variable (si se va a hacer compresión de datos).
- b) estudiar la necesidad de crear algunas variables nuevas y modificar algunas ya existentes. En el caso de Costa Rica, las variables de vivienda sólo se encontraban en el primer registro de hogar de cada vivienda; si una vivienda tenía más de un hogar, los registros 2 en adelante de tipo vivienda tenían esa información en blanco. Para poder procesar luego esa información correctamente en REDATAM era necesario copiar la información de la vivienda a cada registro de hogares. También, era necesario crear dos variables adicionales que facilitarían luego el procesamiento (una que indicaba si la persona había nacido en el mismo lugar de empadronamiento y otra que indicaba si la persona había residido en el mismo lugar de empadronamiento cinco años atrás).
- c) decidir las variables geográficas a utilizar, además de identificar cuáles llevarán nombre en la estructura geográfica de REDATAM. En el caso de Costa Rica se decidió usar cuatro niveles geográficos (Provincia, Cantón, Distrito y Segmento), de los cuales solamente los tres primeros tendrían nombres en REDATAM.

En el caso de Costa Rica, una vez definidos estos puntos, se crearon los siguientes cuadros resúmenes:

Variables de vivienda

Variable	Rango	Tamaño bytes	Tamaño bits	Blanco asigna	Espacio en Mb
Vivienda	0-552	3	10		.664
Hogar	0-9	1	4		.265
Zona urb-rur.	1-4	1	3		.199
Tipo-vivienda	1-5	1	3	0	.199
Tipo-ocupación	1-7	1	3	0	.199
Alquiler	0-99999	5	17	0	1.128
Pared	1-12	2	4	0	.265
Techo	1-12	2	4	0	.265
Piso	1-10	2	4	0	.265
Tot.aposentos	0-24	2	5	0	.332
Dormir	0-10	2	4	0	.265
Comedor	0-3	2	2	0	.133
Cocina	0-3	2	2	0	.133
Cocina-comedor	0-3	2	2	0	.133
Otros Usos	0-5	2	3	0	.199
Abast.agua	1-9	1	4	0	.265

2. LOS DATOS ESTAN EN UN COMPUTADOR CENTRAL -----

Cont.agua ver.	1-5	1	3	0	.199
Cont.agua inv.	1-5	1	3	0	.199
Serv.baño	1-3	1	2	0	.133
Serv.sanitario	1-9	1	4	0	.265
T.alumbrado	1-5	1	3	0	.199
Combustible	1-7	1	3	0	.199
Radio	1-2	1	2	0	.133
Cocina	1-2	1	2	0	.133
Plancha	1-2	1	2	0	.133
TV ByN	1-2	1	2	0	.133
TV color	1-2	1	2	0	.133
Refrigerador	1-2	1	2	0	.133
Lavadora	1-2	1	2	0	.133
Calentador	1-2	1	2	0	.133
Teléfono	1-2	1	2	0	.133
Cepillo electr.	1-2	1	2	0	.133
Tanque agua	1-2	1	2	0	.133
Vehículo	1-2	1	2	0	.133
Finca agro.	1-9	1	4	0	.265
Ganado-vacuno	1-9	1	4	0	.265
Total personas	0-99	2	7	0	.465
Hombres	0-99	2	7	0	.465
Mujeres	0-99	2	7	0	.465

		57	145	TOTAL ESPACIO	9.356 Mb

Variables de Población

Variable	Rango	Tamaño bytes	Tamaño bits	Blanco asignar	Espacio en Mb
Relación c/Jefe	1-9	1	4		1.152
Sexo	1-2	1	2		.576
Edad	0-99	2	7		2.017
Nació aquí	0-1	1	1		.288
Prov.nacimiento	0-7	1	3		.864
Cantón nac.	0-20	2	4		1.152
Distrito y país	0-701	3	10		2.881
Año Llegada	0-99	2	7	0	2.016
Nacionalidad	7-998	3	10	0	2.881
Seguro Social	1-5	1	3		.864
Prov.5 años	1-9	1	4	0	1.152
Cantón 5 años	0-88	2	7	0	2.016
Residía aquí	0-1	1	1		.228
Matr. escolar	1-2	1	2	0	.576
Nivel Instr.	0-39	2	6	0	1.729
Leer y escr.	1-2	1	2	0	.576
Estado Civil	1-6	1	3	0	.864
Tipo Actividad	1-7	1	3	0	.864
Ocupación	0-980	3	10	0	2.881
Cat.ocupación	1-4	1	3	0	.864

2. LOS DATOS ESTAN EN UN COMPUTADOR CENTRAL -----

Rama Actividad	0-9600	4	14	0	4.033
Horas trabaj.	1-80	2	7	0	2.016
Hijos tenidos	0-99	2	7	0	2.016
Hijos sobreviv.	0-99	2	7	0	2.016
		-----	-----		-----
		41	127	TOTAL ESPACIO	36.52 Mb

2.4 Creación del Diccionario de Datos.

Con esa información se debe crear el diccionario de datos para REDATAM. En él se indica el nombre de la base de datos (cuatro caracteres, por ej. CR84 que se usó para Costa Rica), se indican las variables a incluir en la base de datos, su rango, posibles valores y el tipo a que pertenecen (identificación geográfica, población o vivienda), para mayores detalles ver descripción del Diccionario de Datos en Manual del Usuario REDATAM, Apéndice D.

En esta definición del diccionario de datos hay que asignar un nombre a cada variable, y REDATAM le asigna además un número consecutivo, el cual será necesario luego para darle nombre a los archivos correspondientes a cada variable.

2.5 Procesos a realizar en el computador central.

En este punto, es necesario realizar algunos procesos en el computador central, a fin de preparar los archivos en los formatos que REDATAM lo exige para la creación de las bases de datos:

- a) ordenamiento de los datos por las áreas geográficas que se utilizarán en REDATAM. Con el Censo de Costa Rica se utilizó el utilitario SORT del sistema operativo OS/VS1, el cual tardó 12 minutos de CPU en un computador IBM 3031 ordenando 2.973.385 registros por una clave de 15 caracteres. Este proceso normalmente toma mucho tiempo y espacio en disco, lo que implica que en algunos casos deba hacerse por regiones y luego concatenarse en los archivos de salida.
- b) del archivo ordenado deben obtenerse tres archivos, uno con las áreas geográficas a utilizar (archivo de cortes descrito en el punto 2.9), otro con las áreas geográficas que llevarán nombre en REDATAM (archivo de nombres que se describe en el punto 2.7) y otro con el número de personas de cada hogar (archivo de punteros, ver punto 2.10). Para esto se escribió un programa en lenguaje COBOL ("CORTES COBOL x" en CMS, ver apéndice F.1), el cual tardó 4 minutos, generó 10.817 registros de áreas geográficas, 508 de áreas con nombres y 556.776 de punteros (personas por hogar).
- c) se debe generar un archivo por cada variable de vivienda, esto a partir de los datos ordenados. Este archivo se utilizará luego para crear los archivos ".BIN" en REDATAM. Para el censo del ejemplo se

2. LOS DATOS ESTAN EN UN COMPUTADOR CENTRAL -----

creó un programa en lenguaje COBOL que hacía esta tarea ("VIVIENDA COBOL x", ver apéndice F.2). Por la cantidad de archivos que creó, fue necesario correrlo en una partición especial del sistema operativo la que además debía tener suficiente espacio en disco para la grabación de estos archivos. Su ejecución consumió 15 minutos de CPU para generar los 39 archivos de variables de las 556.776 viviendas.

- d) en forma similar se debe crear un archivo por cada una de las variables de población a incluir. Para el censo del ejemplo, se escribió también un programa en lenguaje COBOL ("POBLACION COBOL x", ver apéndice F.3), muy semejante al programa anterior, el cual consumió 36 minutos de CPU generando los 24 archivos correspondientes a cada variable de población (de 2.416.809 personas). En este caso, se deben considerar espacios aún mayores en disco, pues el tamaño de los archivos es mucho mayor que para las viviendas.

Como sugerencia de orden organizativo, es aconsejable que en los procesos anteriores se asigne a las variables el mismo nombre que se les dio en el diccionario de datos de REDATAM.

2.6 Transmisión de los datos.

La transmisión al microcomputador de los archivos generados anteriormente, es una etapa que se puede ir haciendo en forma paralela a la creación de ellos. En el caso de Costa Rica, se utilizaron las noches para transmitir la información y en el día se iban creando los otros archivos. Los detalles propios de la transmisión de la información son muy particulares para cada instalación; en el ejemplo se empleó la tarjeta AST PCOX, utilizando el modo CMS, el cual mostró las siguientes peculiaridades (las que son muy similares entre todas las tarjetas actualmente existentes de transmisión IBM MAINFRAME - IBM PC):

-los archivos debían estar en CMS, por lo que fue necesario copiarlos de cinta o disco del OS al CMS, y en esta copia se aprovechó para darles un bloqueo conveniente.

-permitía transmitir archivos con un largo de registros máximo de 255 caracteres.

-los registros cuyos últimos caracteres son espacios en blanco, eran truncados en el último carácter no blanco.

-la velocidad de transmisión era de alrededor de 30 minutos por megabyte de datos a 9600 baud.

-el método de transmisión involucra la edición del archivo con el XEDIT de CMS, por lo cual la máquina virtual a utilizar debía tener mucha memoria (16 Mb por ejemplo).

2. LOS DATOS ESTAN EN UN COMPUTADOR CENTRAL -----

En el caso de Costa Rica - y bajo estas circunstancias -, para sacarle provecho a la velocidad de transmisión y para aprovechar mejor el espacio en disco, tanto en CMS como en el microcomputador, los archivos se armaron formando registros de 250 caracteres, o el número más próximo que fuera múltiplo del tamaño de la variable (por ej. 240 para variables de 3 bytes); y se grabaron en CMS en bloques de 1000 caracteres (o 960).

La transmisión de la información se efectuó durante las noches, aprovechando la modalidad "batch" que permite la tarjeta PCOX, de manera que se intentaba transmitir todas las variables que entraran en el espacio en disco disponible en el microcomputador. Al día siguiente se respaldaba esta información en el microcomputador y se liberaba el espacio. Otra modalidad que se podría efectuar, que ahorra espacio en los respaldos, es efectuar la compresión de los datos (lo que se explicará más adelante) en cuanto esos archivos sean transmitidos y respaldar solamente el archivo comprimido (llamado ".BIN").

Por la cantidad de variables a transmitir, es necesario llevar un estricto control de este proceso. Para ello se sugiere llevar un formulario donde se anota el nombre de la variable, número en REDATAM, tamaño en bytes, número de registros en el MAINFRAME, fecha de transmisión, tamaño en el microcomputador, número de la unidad de respaldo (diskette o cinta), fecha de la compresión, tamaño del archivo comprimido, y cualquier otra información que se considere relevante. En el Apéndice D, se presenta una muestra de un formulario estándar con estas informaciones.

Para un mayor entendimiento de los pasos en el "mainframe", se presenta un flujo de programas en el Apéndice E.

Después de transmitir todos los archivos, el trabajo se limita al microcomputador. En el caso de Costa Rica, se usó un IBM PC AT con 30 Mb de disco duro de los cuales 20 estaban disponibles para este proceso.

2.7 Actualización del archivo de nombres.

Una vez que se ha creado el archivo de nombres y se ha transmitido al microcomputador, es necesario agregarle manualmente los nombres de las áreas geográficas que llevarán nombre en REDATAM. Esto puede hacerse con cualquier editor de texto o procesador de palabras que produzca un archivo ASCII. Debe calcularse el largo del nombre mayor para cada área, pues éste se utilizará luego en la generación interna del archivo geográfico de REDATAM.

El siguiente es un fragmento del archivo geográfico utilizado para CR84, donde la parte titulada 'xppcdd' fue generada en el 'mainframe' y la parte 'nnnnnnnnnnnnnn...' fue agregada manualmente utilizando WORDPERFECT (saliendo con un archivo ASCII al final):

2. LOS DATOS ESTAN EN UN COMPUTADOR CENTRAL -----

"<base de datos>.COR": archivo de cortes geográficos que contiene un registro por cada área geográfica y el total de viviendas para esa área. Este archivo fue generado en el punto 2.5.b) y transmitido al micro. El formato es CODIGO COMPLETO DEL AREA seguido de la cantidad de VIVIENDAS.

Ejemplo:

```

pccddssstttttttt
|-----|
reg. 1 |1010100100000123| 'p'   : Provincia (1o. área)
reg. 2 |1010100200000101| 'cc'  : Cantón (2o. área)
reg. 3 |1010100300000045| 'dd'  : Distrito (3o. área)
...   |. . . . .| 'sss' : Segmento (4o. área)
|-----| 'ttt..' : Total de hogares del área
    
```

"<base de datos>.NOM": archivo con los nombres de las áreas geográficas que llevarán nombre. El formato de este archivo fue descrito en el punto 2.7.

Con estos archivos listos se debe llamar a REDATAM, seleccionar la opción de "Mantención de la base de datos", luego la opción "Carga Manual" y, finalmente, la opción "Generación de índices". Para la base de datos CR84 este proceso tardó 24 minutos.

En este paso los archivos generados tienen el nombre "bbbbnnnn.INX" donde

```

'bbbb' : nombre de la base de datos
'nnnn' : número de la variable en el diccionario de datos para la
         variable geográfica
    
```

por ejemplo, para la base de datos CR84 se generaron los archivos:

```

CR840001.INX      (índices de Provincia: 1o. nivel geográfico)
CR840002.INX      (índices de Cantón: 2o. nivel geográfico)
CR840003.INX      (índices de Distrito: 3o. nivel geográfico)
CR840004.INX      (índices de Segmento: 4o. nivel geográfico)
    
```

2.10 Generación de enlaces jerárquicos.

Este proceso crea el archivo "<base de datos>PUNT.RED" (por ej. CR84PUNT.RED), el cual contiene punteros para cada vivienda de la base de datos, y también contiene el número de personas de cada vivienda.

Para crear este archivo hay que utilizar la opción "Mantención de la base de datos" de REDATAM, luego la opción "Carga manual", y luego se selecciona "Genera acceso jerárquico"; éste pide el nombre de la base de datos y luego se presiona la tecla "PgDn". En CR84 este proceso tardó 28 minutos.

La entrada a este proceso es el archivo de punteros generado en el punto 2.5.b), su formato es simplemente un registro por cada vivienda (en realidad hogar) del censo y cada registro contiene el número de personas

2. LOS DATOS ESTAN EN UN COMPUTADOR CENTRAL -----

del hogar. Su nombre debe ser "<base de datos>.POB", por ej. "CR84.POB".
Ejemplo de un archivo:

```

reg. 1 |--| (primer hogar con 5 personas)
reg. 2 |02| (segundo hogar con 2 personas)
reg. 3 |11| (tercer hogar con 11 personas)
reg. 4 |00| (cuarto hogar sin personas: vivienda desocupada)
reg. 5 |01| (quinto hogar con una persona)
...   |..| . . . . .
      |--|
    
```

2.11 Generación de archivos invertidos.

Esta es la última etapa del proceso de generación propiamente tal de la base de datos. Aquí se genera un archivo, en formato REDATAM, para cada variable del sistema (excepto las variables geográficas).

Antes de iniciar este proceso se deben tener los archivos correspondientes a cada variable definidos con el nombre "<base de datos>nnnn", donde nnnn son cuatro dígitos correspondientes al número de la variable según se definió en el diccionario de datos. Ejemplos:

```

CR840005      variable 'Número de la vivienda' en CR84
CR840008      variable 'Tipo de vivienda' en CR84
CR840044      variable 'Relación de parentezco con el jefe' en CR84.
    
```

Para realizar este proceso hay que seleccionar la opción "Genera archivos invertidos" de la sección de "Carga manual", éste pide el nombre de la base de datos y el número de la variable a invertir (y comprimir si es el caso). Se le pueden dar varios números de variables para invertir, y una vez que se le han indicado, se presiona la tecla "PgDn" y empieza el proceso.

Al final, se deben haber creado tantos archivos como variables se le dieron, los nombres de estos archivos tienen la forma "<base de datos>nnnn.BIN" y son los archivos de datos finales que deben estar presentes para la ejecución de REDATAM. Ejemplos:

```

CR840005.BIN  archivo invertido y comprimido de No. de viv. en CR84
CR840008.BIN  archivo invertido y comprimido de tipo de viv. en CR84
CR850044.BIN  archivo invertido y comprimido de relación con el jefe.
    
```

El tiempo calculado según el ejemplo de Costa Rica fue de aproximadamente 20 minutos por megabyte.

2. LOS DATOS ESTAN EN UN COMPUTADOR CENTRAL -----

2.12 Verificación del sistema.

La prueba del sistema es una labor que se puede ir llevando a cabo en el transcurso de la instalación. Cada vez que se invierte una variable se pueden ir sacando marginales de ella y comparando con los marginales originales. Para esto es necesario haber creado previamente el archivo "<base de datos>PUNT.RED".

Debido al tiempo que pueden demorar los marginales, se pueden sacar para un área menor, preferiblemente que involucre la última área, pero para esto se requiere tener datos sobre esta área a fin de poder verificar los resultados obtenidos. Sin embargo, al menos para unas variables claves es necesario obtener marginales desde REDATAM para todo el universo. En estas pruebas, es necesario tener a mano las publicaciones censales y todos los tabulados posibles, a fin de comparar resultados. Si se descubre algún problema debe detenerse toda la instalación del sistema hasta llegar a detectar su causa y posibles implicaciones en otras variables.

Utilizando REDATAM en CR84, la distribución de frecuencias del censo completo para una variable de vivienda en el total del país demoró 50 minutos, y para una variable de población, aproximadamente 10 horas.

2.13 Respaldos.

La generación del sistema REDATAM requiere mucho espacio en disco, por eso generalmente es necesario respaldar algunos archivos (su grabación en diskette o cinta) para liberar espacio en disco duro.

2.14 Requisitos finales de espacio.

Una vez que se tiene todo el sistema instalado y probado, los archivos necesarios son bastante menos y más pequeños que los utilizados durante la generación del proceso. De estos archivos definitivos se deben sacar copias de respaldo en la forma usual que se hace con todos los sistemas.

El espacio total ocupado para la base de datos CR84 fue el siguiente:

- Variables de vivienda:	9.5 Mb
- Variables de población:	39.5 Mb
- Variables de índices y diccionario de datos	.2 Mb
- Archivo de punteros	2.2 Mb
 TOTAL EN DISCO	 51.4 Mb

Se insiste en que un requisito indispensable es tener suficientes medios de respaldo (cintas o diskettes de alta densidad).

3. AGREGANDO NUEVAS VARIABLES A UNA BASE DE DATOS YA EXISTENTE -----

3. AGREGANDO NUEVAS VARIABLES A UNA BASE DE DATOS YA EXISTENTE

3.1 Situación 1: La variable puede derivarse de variables ya existentes.

Como ejemplo, supongamos que se desea crear una nueva variable llamada GROUP5, la cual tendrá la recodificación de variable AGE dentro de los grupos de 5 años de edad. Puede calcularse en REDATAM cada vez que sea necesario, pero si se piensa que esto demorará demasiado tiempo y se tiene suficiente espacio en el disco duro como para mantener una variable más, se pueden seguir los siguientes pasos:

3.1.1 Procedimiento inicial.

Antes que nada, se debe estar seguro de tener una copia de respaldo de la base de datos ya existente. Para este ejemplo usaremos la base de datos de Miranda que se encuentra incluida con REDATAM.

3.1.2 Creación de la variable.

Usando un archivo de comando ECF se crea un archivo "rectangular" conteniendo solamente la variable que se desea agregar. Los comandos de REDATAM serían como los siguientes:

```
run name "Append a new variable GROUP5"  
geography REGALL  
compute GROUP5 = (AGE/5) + 1  
var label GROUP5 "Age in 5-year age groups"  
val label GROUP5 1 " 0 - 4 years" 2 " 5 - 9 years" 3 "etc..."  
write GROUP5 (F2.0)  
option filename = MI800058
```

Si se carga estos comandos dentro del procesador estadístico y se ejecutan, ellos crearán un archivo "rectangular" llamado MI800058, conteniendo solamente una variable (GROUP5).

El comando GEOGRAPHY REGALL ha incluido todos los registros de la base de datos. Cuando este se CREA en la etapa de SELECCION GEOGRAFICA de REDATAM, en ese momento hay que poner una "t" en TODAS las áreas geográficas mayores.

La opción "(F2.0)" se usa para forzar a la variable a tener un tamaño de dos caracteres.

El nombre de archivo del archivo "rectangular" depende del esquema de nombres del programa REDATAM donde "MI80" es el nombre de la base de datos y "0058" es la próxima variable disponible en el diccionario REDATAM (si se mira en el diccionario de MIRANDA se verá que hay 57 variables).

3. AGREGANDO NUEVAS VARIABLES A UNA BASE DE DATOS YA EXISTENTE -----

3.1.3 Actualización del diccionario.

Del menú principal de REDATAM, hay que ir a la Opción de Actualización del Diccionario de Datos (letras D, F y A en este mismo orden). Hay que ir al final de las variables existentes y luego crear una nueva variable denominada GROUP5, que sería la número 58, de acuerdo a lo señalado en el punto anterior. En este caso, si se desea comprimirla, hay que ingresar una "b" en el campo tipo de dato. El largo original es 2, la posición inicial es 100, la próxima posición disponible en el archivo "rectangular original", para garantizar que no ocurrirán superposiciones cuando se esté generando un sub-set de esta base de datos con el comando WRITE (la última variable en MIRANDA es INCOME, empezando en la posición 95 y largo original igual 5). Los rangos mínimo y máximo son 1 y 20 respectivamente, y el tipo de registro es 2. Salir de la Actualización salvándola (con <ALT>D).

3.1.4 Generación de la base de datos.

Con la opción de Carga Manual de la función Mantenimiento de Base de Datos, se genera la variable GROUP5 aplicando la opción "Generación de archivos invertidos". Ingresar luego el nombre de la base de datos (MI80) y el número de variable (58). La tecla PgDn comienza el proceso, creando un archivo denominado MI800058.BIN, con los valores para la variable GROUP5.

3.1.5 Comentarios.

Si se desea crear más de una variable, se tiene dos alternativas:

- a) realizar una vez más las etapas antes mencionadas para cada nueva variable que se desea crear;
- b) realizar la generación de la variable solamente una vez, con una instrucción WRITE, la que debe contener todas las variables que se deseen crear y un OPTION FILENAME con un archivo temporal. Escribir un programa, por ejemplo en BASIC, para leer el archivo temporal y crear en él un archivo separado para cada variable con un nombre de archivo de acuerdo con el número de variable en el diccionario. Actualizar el diccionario y generar las nuevas variables en la forma explicada en los puntos anteriores.

3.2 Situación 2: La variable es una nueva variable.

En este caso, el proceso consiste en obtener un archivo "rectangular" con la variable a ser agregada y luego seguir las etapas 3.1.3 y 3.1.4 señaladas más arriba. El usuario tiene que asegurarse que el archivo "rectangular" conteniendo la nueva variable tiene el mismo número de casos que la variable original.

APENDICE A.1 - PROGRAMA GUYCODE.BAS

```

1100 ' recoding program and batch creation
1400 DIM WEIGHT(10,2),ARCS$(150)
1500 FOR I=1 TO 10:FOR J=1 TO 2:READ WEIGHT(I,J):NEXT J:NEXT I
1600 DATA 12,4, 14,8, 19,75, 20,5, 18,6, 19,4, 12,8, 4,4, 8,2, 14,0
1700 FOR I=1 TO 9:READ REMAIN(I):NEXT I
1800 DATA 5,0,4,0,3,0,2,0,1
1900 A$="C:\guyredem\": B$="C:\guyredem\" ' input and output disks
2100 RECLLEN=61:CONTIN=0:CONTOUT=0:CONTOT=0
2200 CLS:PRINT:PRINT "Recoding program":PRINT
2300 INPUT "enter initial region number ";REGINIT
2400 INPUT "enter final region number ";REGFINA
2600 PRINT:INPUT "enter output file name (without extension) ";M$
2700 OPEN "o",#2,B$+M$: OPEN "o",#3,B$+M$+".LST"
2900 FOR JJ = REGINIT TO REGFINA
3000 SEED=0
3100 IF JJ = 3 THEN MAXSEED=4:MAXCOUNT=1: GOTO 3400
3200 IF JJ = 4 THEN MAXSEED=2:MAXCOUNT=1: GOTO 3400
3300 MAXSEED=5:MAXCOUNT=REMAIN(WEIGHT(JJ,2)+1)
3400 IF JJ<10 THEN M$="0"+RIGHT$(STR$(JJ),1) ELSE M$=RIGHT$(STR$(JJ),2)
3500 SHELL "DIR "+M$+"*.FOR > DIR.LST"
3600 OPEN "I",#1,"DIR.LST": I=0
3700 WHILE NOT EOF(1)
3800 INPUT #1,MM$: IF LEFT$(MM$,2)<>M$ THEN 4000
3900 I=I+1: ARCS$(I)=MM$
4000 WEND: MAXARC=I: CLOSE #1
4100 FOR I=1 TO MAXARC
4200 CONTIN=0
4300 OPEN "r",#1,A$+LEFT$(ARCS$(I),7)+".for",RECLLEN:CONTIN=0
4400 FIELD #1,RECLLEN AS RECO$: LASTREC=LOF(1)/RECLLEN
4500 PRINT "reading file ";LEFT$(ARCS$(I),7);
4600 CONTIN=CONTIN+1
4700 IF CONTIN>LASTREC THEN CONTOT=CONTOT+CONTIN-1:CLOSE #1:GOTO 6900
4800 GET #1,CONTIN: ENTR$=RECO$: TYPEREC=VAL(LEFT$(ENTR$,1))
4900 IF TYPEREC <> 1 THEN 5200
5000 INTER$=MID$(ENTR$,13,4) 'saves date of interview
5100 GOSUB 10200 ' weight definition
5200 ' adds date of interview and weight to every record
5300 ENTR$=ENTR$+INTER$+WEIGHT$
5400 IF TYPEREC <> 2 THEN ENTR$="": GOTO 4600
5500 ' age and age group calculations
5600 AGE = 86 - VAL(MID$(ENTR$,21,2))
5650 MON1$ = MID$(ENTR$,19,2): MON2$ = MID$(ENTR$,64,2)
5660 DAY1$ = MID$(ENTR$,17,2): DAY2$ = MID$(ENTR$,62,2)
5700 IF MON1$ > MON2$ THEN AGE=AGE-1
5800 IF MON1$ = MON2$ AND DAY1$ > DAY2$ THEN AGE=AGE-1
5850 A$ = STR$(AGE)

```

```
5900     IF AGE < 10 THEN AGE$="0"+RIGHT$(A$,1) ELSE AGE$=RIGHT$(A$,2)
6000     IF AGE > 85 THEN AGE=85
6100     GROUP=INT(AGE/5)+1: A$=STR$(GROUP)
6200     IF GROUP<10 THEN GROUP$="0"+RIGHT$(A$,1) ELSE GROUP$=RIGHT$(A$,2)
6300     ' adds calculated age and age group to person records
6400     ENTR$=ENTR$+AGE$+GROUP$: CONTIN(TYPEREC)=CONTIN(TYPEREC)+1
6700     CONTOUT(TYPEREC)=CONTOUT(TYPEREC)+1
6750     PRINT #2,ENTR$:CONTTOT=CONTTOT+1: ENTR$="": GOTO 4600
6900     PRINT #3,ARCS$(I);" records read =";
7060     PRINT #3,USING "#####";CONTIN - 1
7100     NEXT I: PRINT
7300     PRINT #3,"total input records for region ";JJ;" = ";
7450     PRINT #3,USING "#####";CONTTOT
7500     FOR I=1 TO 10
7600         IF CONTIN(I)=0 THEN 8000
7800         PRINT #3," type ";I;" = ";;PRINT #3,USING "#####";CONTIN(I)
7900         CONTIN(I)=0
8000     NEXT I:PRINT #3,""
8200     PRINT #3,"total output records for region ";JJ;" = ";
8250     PRINT #3,USING "#####";CONTTOT
8300     FOR I=1 TO 10
8400         IF CONTOUT(I)=0 THEN 8900
8500         FINAL(I)=FINAL(I)+CONTOUT(I)
8750         PRINT #3,USING "#####";CONTOUT(I): CONTOUT(I)=0
8900     NEXT I:PRINT "":PRINT #3,"":CONTTOT=0:CONTTOT=0
9000     NEXT JJ
9100     PRINT #3,"":PRINT #3,"final results":PRINT #3,""
9200     FOR I = 1 TO 10
9300         IF FINAL(I)=0 THEN 9700
9500         PRINT #3," type ";I;" = ";;PRINT #3,USING "#####";FINAL(I)
9600         FINAL=FINAL+FINAL(I)
9700     NEXT I:PRINT #3,""
9900     PRINT #3," total = ";;PRINT #3,USING "#####";FINAL
9950     CLOSE #3, #2:PRINT "":SYSTEM: END
10200 ' weight definition
10300     SEED=SEED+1:WEIGHT=WEIGHT(JJ,1)
10400     IF SEED > MAXSEED THEN SEED = 1:GOTO 10600
10500     IF SEED > MAXCOUNT THEN WEIGHT=WEIGHT+1
10550     A$=STR$(WEIGHT)
10600     IF WEIGHT<10 THEN WEIGHT$="0"+RIGHT$(A$,1) ELSE WEIGHT$=RIGHT$(A$,2)
10700     RETURN
```

APENDICE A.2 - PROGRAMA GUYNOM.BAS

```
1000 ' guynom.bas - creates the gy86.nom file and checks sort sequence
1100 NNAME$(1)="REGION": NNAME$(2)="MAJOR AREA": NNAME$(3)="ED"
1250 CLS:PRINT "Guynom Program - creates GY86.NOM":PRINT
1300 OPEN "I",#1,"GY86": OPEN "O",#2,"GY86.NOM": OPEN "O",#3,"GY86NOM.LST"
1450 INPUT #1,ENTR$:TIPO1=1:AREANT$=MID$(ENTR$,2,7)
1500 WHILE NOT EOF(1)
1600 INPUT #1,ENTR$:TYPEREC=VAL(LEFT$(ENTR$,1)):AREAB$=MID$(ENTR$,2,7)
1650 IF TYPEREC <> 1 THEN 2500
1700 TIPO1=TIPO1+1
1750 IF AREAB$= AREANT$ GOTO 2700
1800 IF AREAB$> AREANT$ THEN 1950
1850 PRINT "sequence error ";AREAB$;" ";AREANT$
1900 PRINT #3,"sequence error ";AREAB$;" ";AREANT$
1950 CODE$(1)= MID$(AREANT$,1,2): CODE$(2)= MID$(AREANT$,3,2)
2050 CODE$(3)= MID$(AREANT$,5,3)
2100 IF MID$(AREAB$,1,2) = CODE$(1) THEN 2250
2150 FOR I=1 TO 3:GOSUB 3350:NEXT I: GOTO 2400
2250 IF MID$(AREAB$,3,2) = CODE$(2) THEN 2350
2300 I=2:GOSUB 3350: I=3:GOSUB 3350: AREANT$ = AREAB$: GOTO 2700
2500 TIPO2=TIPO2+1
2510 IF AREAB$=AREANT$ THEN 2700
2550 PRINT #3,"type 2 does not have the same id as type 1 ";ENTR$
2600 PRINT #3,"type 2 does not have the same id as type 1 ";ENTR$
2650 GOTO 1950
2700 WEND
2750 CODE$(1)= MID$(AREANT$,1,2): CODE$(2)= MID$(AREANT$,3,2)
2850 CODE$(3)= MID$(AREANT$,5,3)
2900 FOR I=1 TO 3:GOSUB 3350:NEXT I
2950 PRINT:PRINT "records type 1 = ";TIPO1
3000 PRINT " type 2 = ";TIPO2
3050 PRINT " total = ";TIPO1+TIPO2
3100 PRINT #3,"":PRINT #3,"records type 1 = ";TIPO1
3150 PRINT #3," type 2 = ";TIPO2
3200 PRINT #3," total = ";TIPO1+TIPO2
3250 CLOSE #1,#2,#3: SYSTEM: END
3350 ' areabreaks
3400 SAL$="0"+RIGHT$(STR$(I),1)+CODE$(1)
3450 IF I=1 THEN SAL$=SAL$+"00000":GOTO 3650
3500 SAL$=SAL$+CODE$(2)
3550 IF I=2 THEN SAL$=SAL$+"000":GOTO 3650
3600 SAL$=SAL$+CODE$(3): SAL$=SAL$+NNAME$(I)+" "+CODE$(I)
3700 PRINT #2,SAL$: RETURN
```

APENDICE B - ESTIMACION DEL ESPACIO PARA LA BASE DE DATOS

Hay que hacer estimaciones para cuatro tipos de datos: variables de vivienda, variables de población, índices geográficos y un archivo adicional que va a contener la relación entre viviendas y sus ocupantes (archivo de punteros).

B.1 - Variables de vivienda y de población.

El método de cálculo para las variables de vivienda y población es el mismo, variando sólo el multiplicador final, que en el caso de vivienda es el número de viviendas, y para las variables de población es el número de personas. Si las variables no van a ser comprimidas, el proceso es muy sencillo: se suma el número de bytes ocupados por cada tipo de variable (vivienda y población), sin contar con las variables de identificación geográfica, y se multiplican estos números por el número de registros de cada caso.

Para una base de datos comprimida, el espacio total comprimido depende del sistema de compresión utilizado. Hasta el momento, tomando en cuenta que REDATAM trabaja solamente con variables numéricas, el único método que se ha usado es el de cambiar el formato interno de las variables de ASCII en binario. En resumen, una variable que ocupe un byte, con valores de 0 a 9, será almacenada en 4 bits (que puede representar de 0 a 15). Si esta misma variable de un byte tuviera un valor máximo de 7 en vez de 9, ella podría almacenarse en sólo 3 bits (que puede representar de 0 a 7).

El proceso se resume en determinar el espacio en bits ocupado por las variables de vivienda y de población separadamente, dividirlos por 8 (un byte tiene 8 bits), y multiplicarlos por el número de viviendas y de población. La tabla siguiente contiene el número de bits necesarios para cada valor máximo de la variable.

Valor máximo	1	3	7	15	31	63	127	255	511	1023	2047	...
Tamaño en bits	1	2	3	4	5	6	7	8	9	10	11	...
Valor máximo	4095	8191	16383	32767	65535	131071	262143	524287	1048575	2097151	4194303	...
Tamaño en bits	12	13	14	15	16	17	18	19	20	21	22	...

La fórmula consiste en contar el número de variables que ocuparán 1 bit, 2 bits, etc, y multiplicarlos por el respectivo tamaño en bits, luego sumar los totales obtenidos en dos grupos (vivienda y población), dividir estos totales por 8, y luego multiplicarlos por el número de casos (viviendas y personas).

$$V = \sum ((V_i * i) / 8) * NUMVIV.$$

$$P = \sum ((P_i * i) / 8) * NUMPOB.$$

Vi es el número de variables de vivienda con tamaño i en bits,
Pi es el número de variables de población con tamaño i en bits,
NUMVIV es el número de viviendas, y
NUMPOB es el número de personas.

Para Costa Rica, de acuerdo al cuestionario del censo, tenemos:

Número de variables que ocuparán <u>i</u> bits		
i	Vivienda	Población
1		2
2	16	3
3	8	5
4	8	3
5	1	
6		1
7	3	6
10	1	3
14		1
17	1	
casos	550.000	2.500.000

Aplicándose la fórmula se obtiene $V = 9.6 \text{ Mb}$ y $P = 39.6 \text{ Mb}$.

B.2 - Índices geográficos.

El tamaño total de estos archivos (uno para cada índice) depende del total de valores de los índices geográficos con nombres, y del total de valores de índices sin nombres. Los que tienen nombre necesitan 37 bytes, mientras que los que no tienen nombre necesitan 7 bytes. Estos números son aproximados porque dependen del tamaño de cada código y de cada nombre.

$$I = \text{CON} * 37 + \text{SIN} * 7.$$

En el caso de Costa Rica habían tres niveles geográficos con nombre y sólo uno sin nombre. La suma de los tres primeros daba como resultado alrededor de 500 registros, mientras que el último nivel, sin nombre, tenía cerca de 10.000 registros. En este caso,

$$I = 500 * 37 + 10.000 * 7 = .1 \text{ Mb}.$$

B.3 - Archivo de punteros.

Este archivo necesita 4 bytes para cada vivienda, es decir,

$$A = \text{NUMVIV} * 4.$$

En el caso de Costa Rica, $A = 550.000 * 4 = 2.2 \text{ Mb}$.

B.4 - Total.

$$T = V + P + I + A.$$

Para Costa Rica, eso significó, aproximadamente, 51.5 Mb.

Al compararse los totales, se verifica que el espacio destinado a los índices geográficos (.1 Mb) es insignificante para grandes bases de datos, lo que no ocurre para pequeñas encuestas, ya que la jerarquía geográfica de un país es la misma tanto para el censo como para una encuesta a nivel nacional.

APENDICE C - ARCHIVOS DEL SISTEMA

<u>NOMBRE</u>	<u>DESCRIPCION</u>	<u>LUGAR DE CREACION</u>
<b.d.>.NOM	archivo con códigos de las áreas geográficas que llevan nombre y su nombre respectivo	MAINFRAME y modificación en MICRO con un editor para poner los nombres
<b.d.>.COR	archivo de cortes geográficos con el total de viviendas en cada área geográfica	MAINFRAME
<b.d.>.POB	archivo de punteros, con un registro por vivienda con el total de personas	MAINFRAME
<b.d.>nmmn	archivos de datos correspondientes a cada una de las variables de datos (no geográficas). 'nmmn' es el número de la variable en el diccionario REDATAM	MAINFRAME
<b.d.>.DIC	archivo creado en la definición del diccionario de datos	REDATAM: Diccionario de datos
<b.d.>.CTL	archivo de control creado en la definición del diccionario	REDATAM: Diccionario de datos
<b.d.>.GEO	archivo geográfico, contiene los nombres y otra información de las variables geográficas	REDATAM: Creación del archivo geográfico
<b.d.>PUNT.RED	archivo de punteros, indica la posición de cada vivienda y la cantidad de personas en ellas	REDATAM: CARGA MANUAL 'Genera acceso jerárquico'
<b.d.>nmmn.INX	archivos de índices, uno por cada área geográfica. 'nmmn' es el número de la variable en el diccionario REDATAM	REDATAM: CARGA MANUAL 'Generación de índices'
<b.d.>nmmn.BIN	archivos de datos propiamente, uno por cada variable. 'nmmn' es el número de la variable en el diccionario REDATAM	REDATAM: CARGA MANUAL 'Genera archivos invertidos'

 APENDICE F.1 - PROGRAMA CORTES

```

//STEP1 EXEC COBUCLG                                COR00010
  IDENTIFICATION DIVISION.                          COR00020
  PROGRAM-ID.    CORTES.                             COR00030
  AUTHOR.        HARRY HERNANDEZ.                   COR00040
  DATE-WRITTEN.  8 MAYO 1987.                       COR00050
  DATE-COMPILED.                               COR00060
  REMARKS.      -----COR00070
                ! ESTE PROGRAMA PERTENECE A LA INSTALACION DEL SISTEMA !COR00080
                ! REDATAM PARA EL CENSO DE COSTA RICA DE 1984.          !COR00090
                ! SU FUNCION ES TOMAR DE LAS CINTAS DE DICHO CENSO      !COR00100
                ! LA IDENTIFICACION GEOGRAFICA Y GENERAR TRES ARCHIVOS !COR00110
                ! 1-ARCHIVO DE CORTES: CONTIENE UN REGISTRO POR CADA    !COR00120
                !   IDENTIFICACION GEOGRAFICA DIFERENTE, EL CUAL      !COR00130
                !   CONTIENE EL AREA GEOGRAFICA Y EL NUMERO DE HOGA- !COR00140
                !   RES DE ESA AREA.                                    !COR00150
                ! 2-ARCHIVO DE NOMBRES: CONTIENE UN REGISTRO POR CADA !COR00160
                !   AREA GEOGRAFICA QUE LLEVARA NOMBRE EN EL SISTEMA !COR00170
                !   REDATAM, EL CONTENIDO DE CADA REGISTRO SERA      !COR00180
                !   LA IDENTIFICACION DE CADA AREA.                  !COR00190
                ! 3-ARCHIVO DE PUNTEROS: UN ARCHIVO CUYO CONTENIDO ES !COR00200
                !   EL NUMERO DE PERSONAS QUE CONTIENE CADA HOGAR.   !COR00210
                ! ----- !COR00220
                ! PARA UTILIZAR ESTE PROGRAMA CON OTROS DATOS ES NECE- !COR00230
                ! SARIO MODIFICAR LOS SIGUIENTES PUNTOS:             !COR00240
                ! * DISEÑO DE REGISTRO DEL CENSO O LA ENCUESTA:      !COR00250
                !   HAY QUE CAMBIAR LA ESTRUCTURA 'REGISTRO-CENSO' !COR00260
                !   ASI COMO LOS DATOS DEL 'FD CENSO'                !COR00270
                ! * ARCHIVO DE CORTES:                                !COR00280
                !   HAY QUE CAMBIAR EL NUMERO Y TAMANO DE LAS VARIA- !COR00290
                !   BLES CORRESPONDIENTES A LAS AREAS GEOGRAFICAS    !COR00300
                !   EN LA ESTRUCTURA 'REGISTRO-CORTES',             !COR00310
                !   ASI COMO LOS DATOS DEL 'FD CORTES'               !COR00320
                ! * ARCHIVO DE NOMBRES:                                !COR00330
                !   CAMBIAR EL NUMERO Y TAMANO DE LAS VARIABLES      !COR00340
                !   GEOGRAFICAS QUE LLEVARAN NOMBRE, POR LO QUE HAY !COR00350
                !   QUE MODIFICAR LA ESTRUCTURA 'REGISTRO-NOMBRES. !COR00360
                ! * LA LOGICA DEL PROGRAMA SUPONE QUE TODA LA IDENTIF. !COR00370
                !   DE CADA REGISTRO SE ENCUENTRA EN POSICIONES CONSE- !COR00380
                !   CUTIVAS (ESTRUCTURAS 'IDENTIFICACION' E         !COR00390
                !   'IDENTIFICACION-ANT', POR LO QUE SERIA NECESARIO !COR00400
                !   MODIFICAR ESTA LOGICA SI NO SE CUMPLE ESA CONDICION !COR00410
                ! * SE SUPONE QUE EL TIPO DE REGISTRO '1' CORRESPONDE !COR00420
                !   A HOGAR Y EL TIPO DE REGISTRO '2' A PERSONA     !COR00430
                ! * EL PARRAFO 'GRABA-NOMBRES' DEBE MODIFICARSE PARA !COR00440
                !   QUE GENERE REGISTROS DEL ARCHIVO DE NOMBRES PARA !COR00450
                !   CADA NIVEL CON NOMBRES, ESTE EJEMPLO PRODUCE     !COR00460
                !   REGISTROS HASTA EL TERCER NIVEL GEOGRAFICO      !COR00470
                !                                                    !COR00480
                ! ----- !COR00490
                ! EN ESTE CENSO LAS VARIABLES GEOGRAFICAS SON:      !COR00500
  
```

```

!      PROVINCIA : NIVEL 1 (CON NOMBRE EN REDATAM) !COR00510
!      CANTON    : NIVEL 2  "  "  "  "          !COR00520
!      DISTRITO  : NIVEL 3  "  "  "  "          !COR00530
!      SEGMENTO  : NIVEL 4 (SIN NOMBRE EN REDATAM) !COR00540
-----COR00550
ENVIRONMENT DIVISION.                                COR00560
CONFIGURATION SECTION.                               COR00570
SOURCE-COMPUTER. IBM-370.                            COR00580
OBJECT-COMPUTER. IBM-370.                            COR00590
INPUT-OUTPUT SECTION.                               COR00600
FILE-CONTROL.                                       COR00610
    SELECT CENSO      ASSIGN TO UT-3410-S-CENSO.      COR00620
    SELECT CORTES     ASSIGN TO DA-3340-S-CORTES.     COR00630
    SELECT NOMBRES    ASSIGN TO DA-3340-S-NOMBRES.    COR00640
    SELECT PUNTEROS   ASSIGN TO DA-3340-S-PUNTEROS.   COR00650
DATA DIVISION.                                       COR00660
FILE SECTION.                                        COR00670
FD CENSO                                             COR00680
    BLOCK CONTAINS 0 RECORDS                          COR00690
    LABEL RECORDS ARE STANDARD                        COR00700
    RECORD CONTAINS 70 CHARACTERS.                   COR00710
* ----- DEBE MODIFICARSE AL LARGO DEL REGISTRO DEL CENSO PARTIC. COR00720
01 REG-CENSO          PIC X(70).                     COR00730
*                                                     COR00740
* ----- LOS BLOQUEOS DE LOS SGTES ARCHIVOS SE HAN CALCULADO COR00750
* PARA DISCOS 3350 PERO PUEDEN MODIFICARSE SEGUN CONVENGA COR00760
FD CORTES                                             COR00770
    LABEL RECORDS ARE STANDARD                        COR00780
    BLOCK CONTAINS 1000 RECORDS                       COR00790
    RECORD CONTAINS 16 CHARACTERS.                   COR00800
* NIVEL GEOGRAFICO 1: 1 CARACTERES                   COR00810
* NIVEL GEOGRAFICO 2: 2 CARACTERES                   COR00820
* NIVEL GEOGRAFICO 3: 2 CARACTERES                   COR00830
* NIVEL GEOGRAFICO 4: 3 CARACTERES                   COR00840
* TOTAL HOGARES      : 8 CARACTERES                   COR00850
* ----- COR00860
* TOTAL 16 CARACTERES                                COR00870
01 REG-CORTES          PIC X(16).                     COR00880
*                                                     COR00890
FD NOMBRES                                           COR00900
    LABEL RECORDS ARE STANDARD                        COR00910
    BLOCK CONTAINS 2000 RECORDS                       COR00920
    RECORD CONTAINS 7 CHARACTERS.                   COR00930
* IDENTIFICADOR DEL NIVEL 2 CARACTERES               COR00940
* NIVEL GEOGRAFICO 1: 1 CHARACTER                    COR00950
* NIVEL GEOGRAFICO 2: 2 CARACTERES                   COR00960
* NIVEL GEOGRAFICO 3: 2 CARACTERES                   COR00970
* TOTAL 7 CARACTERES                                  COR00980
01 REG-NOMBRES          PIC X(07).                     COR00990
*                                                     COR01000
FD PUNTEROS                                           COR01010
    LABEL RECORDS ARE STANDARD                        COR01020
    BLOCK CONTAINS 8000 RECORDS                       COR01030
  
```

	RECORD CONTAINS	2 CHARACTERS.	COR01040
01	REG-PUNTEROS	PIC 9(2).	COR01050
*			COR01060
	WORKING-STORAGE SECTION.		COR01070
77	TOTAL-CENSO	PIC 9(8) COMP VALUE ZERO.	COR01080
77	TOTAL-CORTES	PIC 9(8) COMP VALUE ZERO.	COR01090
77	TOTAL-NOMBRES	PIC 9(8) COMP VALUE ZERO.	COR01100
77	TOTAL-PUNTEROS	PIC 9(8) COMP VALUE ZERO.	COR01110
77	CUENTA-PERSONAS	PIC 9(6) VALUE ZERO.	COR01120
*			COR01130
01	REGISTRO-CENSO.		COR01140
	03 IDENTIFICACION.		COR01150
	05 SEGMENTO.		COR01160
	07 PROVINCIA	PIC X.	COR01170
	07 CANTON	PIC XX.	COR01180
	07 DISTRITO	PIC XX.	COR01190
	07 FILLER	PIC XXX.	COR01200
	05 VIVIENDA-HOGAR	PIC XXX.	COR01210
	05 HOGAR	PIC X.	COR01220
	03 TIPO-REG	PIC X.	COR01230
	03 FILLER	PIC X(57).	COR01240
*			COR01250
01	REGISTRO-CORTES.		COR01260
	03 SEGMENTO-CORTES	PIC X(8).	COR01270
	03 HOGARES-CORTES	PIC 9(8) VALUE 0.	COR01280
*			COR01290
01	REGISTRO-NOMBRES.		COR01300
	03 NIVEL-NOMBRES	PIC XX.	COR01310
	03 PROVINCIA-NOMBRES	PIC X.	COR01320
	03 CANTON-NOMBRES	PIC XX.	COR01330
	03 DISTRITO-NOMBRES	PIC XX.	COR01340
*			COR01350
01	OTRAS-VARS.		COR01360
	03 FIN-ARCH	PIC 9 VALUE 0.	COR01370
	88 FIN-ARCHIVO	VALUE 1.	COR01380
	03 IDENTIFICACION-ANT.		COR01390
	05 SEGMENTO-ANT.		COR01400
	07 PROVINCIA-ANT	PIC X.	COR01410
	07 CANTON-ANT	PIC XX.	COR01420
	07 DISTRITO-ANT	PIC XX.	COR01430
	07 FILLER	PIC XXX.	COR01440
	05 VIVIENDA-ANT	PIC XXX.	COR01450
	05 HOGAR-ANT	PIC X.	COR01460
*			COR01470
	PROCEDURE DIVISION.		COR01480
	OPEN INPUT CENSO, OUTPUT CORTES, NOMBRES, PUNTEROS.		COR01490
	READ CENSO INTO REGISTRO-CENSO AT END MOVE 1 TO FIN-ARCH.		COR01500
	Merge SPACES TO IDENTIFICACION-ANT.		COR01510
	PERFORM GRABA-NOMBRES.		COR01520
	Merge IDENTIFICACION TO IDENTIFICACION-ANT.		COR01530
	PERFORM CICLO THRU CICLO-FIN UNTIL FIN-ARCHIVO.		COR01540
	DISPLAY " TOTAL REGISTROS LEIDOS ", TOTAL-CENSO.		COR01550
	DISPLAY " TOTAL REGISTROS CORTES ", TOTAL-CORTES.		COR01560


```
ADD 1 TO TOTAL-NOMBRES COR02100
MOVE SPACES TO DISTRITO-ANT. COR02110
* ---- ESTE MOVE OBLIGA A GRABAR LUEGO EL DISTRITO COR02120
IF DISTRITO NOT = DISTRITO-ANT COR02130
MOVE "03" TO NIVEL-NOMBRES COR02140
MOVE DISTRITO TO DISTRITO-NOMBRES COR02150
WRITE REG-NOMBRES FROM REGISTRO-NOMBRES COR02160
ADD 1 TO TOTAL-NOMBRES. COR02170
GRABA-PUNTEROS. COR02180
MOVE CUENTA-PERSONAS TO REG-PUNTEROS. COR02190
WRITE REG-PUNTEROS. COR02200
ADD 1 TO TOTAL-PUNTEROS. COR02210
MOVE 0 TO CUENTA-PERSONAS. COR02220
FIN-PROGRAMA. COR02230
EXIT. COR02240
//GO.SYSUDUMP DD SYSOUT=A COR02250
//SYSOUT DD SYSOUT=A COR02260
//CENSO DD UNIT=TAPE, COR02270
// VOL=SER=(R402,R403,R404,R405,R406),DSN=CR84CENS.ORIGIN80 COR02280
//NOMBRES DD UNIT=3350, COR02290
// VOL=SER=OSWORK, COR02300
// DSN=CR84CENS.NOMBRES, COR02310
// DCB=(RECFM=FB,LRECL=7,BLKSIZE=14000), COR02320
// SPACE=(TRK,(10,30),RLSE),DISP=(OLD) COR02330
//PUNTEROS DD UNIT=3350, COR02340
// VOL=SER=OSWORK, COR02350
// DSN=CR84CENS.PUNTERO, COR02360
// DCB=(RECFM=FB,LRECL=2,BLKSIZE=16000), COR02370
// SPACE=(TRK,(10,30),RLSE),DISP=(OLD) COR02380
//CORTES DD UNIT=3350, COR02390
// VOL=SER=OSWORK, COR02400
// DSN=CR84CENS.CORTES, COR02410
// DCB=(RECFM=FB,LRECL=16,BLKSIZE=16000), COR02420
// SPACE=(TRK,(100,100),RLSE),DISP=(OLD) COR02430
```

 APENDICE F.2 - PROGRAMA VIVIENDA

```
//STEP1 EXEC COBUCLG                                VIV00010
  IDENTIFICATION DIVISION.                            VIV00020
  PROGRAM-ID.  VIVIENDA.                               VIV00030
  AUTHOR.      HARRY HERNANDEZ.                       VIV00040
  DATE-WRITTEN. 12 MAYO 1987.                         VIV00050
  DATE-COMPILED.                               VIV00060
  REMARKS.     -----VIV00070
                ! ESTE PROGRAMA PERTENECE A LA INSTALACION DEL SISTEMA !VIV00080
                ! REDATAM PARA EL CENSO DE COSTA RICA DE 1984.           !VIV00090
                ! SU FUNCION ES DIVIDIR LOS REGISTROS DE VIVIENDA (EN    !VIV00100
                ! REALIDAD HOGARES) DE DICHO CENSO Y SEPARAR CADA      !VIV00110
                ! VARIABLE EN UN ARCHIVO DIFERENTE, O SEA, SU SALIDA    !VIV00120
                ! SERAN TANTOS ARCHIVOS COMO VARIABLES DE VIVIENDA     !VIV00130
                ! TENGA EL CENSO.  OTRAS FUNCIONES SERAN LAS DE CONTAR !VIV00140
                ! EL NUMERO DE HOMBRES Y MUJERES PARA GENERAR LAS      !VIV00150
                ! VARIABLES 'TOTAL PERSONAS', 'TOTAL HOMBRES' Y        !VIV00160
                ! 'TOTAL MUJERES DE CADA HOGAR'; TAMBIEN DEBERA COPIAR !VIV00170
                ! LA INFORMACION DE LOS REGISTROS DEL PRIMER HOGAR DE  !VIV00180
                ! CADA VIVIENDA QUE NO SE DIGITO PARA EL RESTO DE LOS  !VIV00190
                ! HOGARES.                                             !VIV00200
                !                                                       !VIV00210
                !                                                       !VIV00220
                -----VIV00230
  ENVIRONMENT DIVISION.                               VIV00240
  CONFIGURATION SECTION.                             VIV00250
  SOURCE-COMPUTER.  IBM-370.                          VIV00260
  OBJECT-COMPUTER.  IBM-370.                          VIV00270
  INPUT-OUTPUT SECTION.                              VIV00280
  FILE-CONTROL.                                       VIV00290
    SELECT CENSO      ASSIGN TO UT-3410-S-CENSO.      VIV00300
*  ----SIGUE LA DEFINICION DE UN ARCHIVO PARA C/VARIABLE DE VIV.  VIV00310
    SELECT VIV        ASSIGN TO DA-3340-S-VIV.        VIV00320
    SELECT HOG        ASSIGN TO DA-3340-S-HOG.        VIV00330
    SELECT V01        ASSIGN TO DA-3340-S-V01.        VIV00340
    SELECT V02A       ASSIGN TO DA-3340-S-V02A.       VIV00350
    SELECT V02B       ASSIGN TO DA-3340-S-V02B.       VIV00360
    SELECT V03        ASSIGN TO DA-3340-S-V03.        VIV00370
    SELECT V04        ASSIGN TO DA-3340-S-V04.        VIV00380
    SELECT V05        ASSIGN TO DA-3340-S-V05.        VIV00390
    SELECT V06A       ASSIGN TO DA-3340-S-V06A.       VIV00400
    SELECT V06B       ASSIGN TO DA-3340-S-V06B.       VIV00410
    SELECT V06C       ASSIGN TO DA-3340-S-V06C.       VIV00420
    SELECT V06D       ASSIGN TO DA-3340-S-V06D.       VIV00430
    SELECT V06E       ASSIGN TO DA-3340-S-V06E.       VIV00440
    SELECT V06F       ASSIGN TO DA-3340-S-V06F.       VIV00450
    SELECT V07        ASSIGN TO DA-3340-S-V07.        VIV00460
    SELECT V08A       ASSIGN TO DA-3340-S-V08A.       VIV00470
    SELECT V08B       ASSIGN TO DA-3340-S-V08B.       VIV00480
    SELECT V09        ASSIGN TO DA-3340-S-V09.        VIV00490
    SELECT V10        ASSIGN TO DA-3340-S-V10.        VIV00500
```

SELECT V11	ASSIGN TO DA-3340-S-V11.	VIV00510
SELECT V12	ASSIGN TO DA-3340-S-V12.	VIV00520
SELECT V13A	ASSIGN TO DA-3340-S-V13A.	VIV00530
SELECT V13B	ASSIGN TO DA-3340-S-V13B.	VIV00540
SELECT V13C	ASSIGN TO DA-3340-S-V13C.	VIV00550
SELECT V13D	ASSIGN TO DA-3340-S-V13D.	VIV00560
SELECT V13E	ASSIGN TO DA-3340-S-V13E.	VIV00570
SELECT V13F	ASSIGN TO DA-3340-S-V13F.	VIV00580
SELECT V13G	ASSIGN TO DA-3340-S-V13G.	VIV00590
SELECT V13H	ASSIGN TO DA-3340-S-V13H.	VIV00600
SELECT V13I	ASSIGN TO DA-3340-S-V13I.	VIV00610
SELECT V13J	ASSIGN TO DA-3340-S-V13J.	VIV00620
SELECT V13K	ASSIGN TO DA-3340-S-V13K.	VIV00630
SELECT V13L	ASSIGN TO DA-3340-S-V13L.	VIV00640
SELECT FINCA	ASSIGN TO DA-3340-S-FINCA.	VIV00650
SELECT GANADO	ASSIGN TO DA-3340-S-GANADO.	VIV00660
SELECT TOTALP	ASSIGN TO DA-3340-S-TOTALP.	VIV00670
SELECT TOTALH	ASSIGN TO DA-3340-S-TOTALH.	VIV00680
SELECT TOTALM	ASSIGN TO DA-3340-S-TOTALM.	VIV00690
*		VIV00700
DATA DIVISION.		VIV00710
FILE SECTION.		VIV00720
FD CENSO		VIV00730
BLOCK CONTAINS 0 RECORDS		VIV00740
LABEL RECORDS ARE STANDARD		VIV00750
RECORD CONTAINS 70 CHARACTERS.		VIV00760
*		VIV00770
* ----- DEBE MODIFICARSE AL LARGO DEL REGISTRO DEL CENSO PARTIC.		VIV00780
*		VIV00790
01 REG-CENSO PIC X(70).		VIV00800
*		VIV00810
*SIGUE LA DEFINICION DE ARCHIVOS PARA CADA VARIABLE DE VIVIENDA		VIV00820
*		VIV00830
FD VIV LABEL RECORD STANDARD BLOCK 6000. 01 RVIV PIC X(3).		VIV00840
FD HOG LABEL RECORD STANDARD BLOCK 19000. 01 RHOG PIC X(1).		VIV00850
FD V01 LABEL RECORD STANDARD BLOCK 19000. 01 RV01 PIC X(1).		VIV00860
FD V02A LABEL RECORD STANDARD BLOCK 19000. 01 RV02A PIC X(1).		VIV00870
FD V02B LABEL RECORD STANDARD BLOCK 3000. 01 RV02B PIC X(5).		VIV00880
FD V03 LABEL RECORD STANDARD BLOCK 8500. 01 RV03 PIC X(2).		VIV00890
FD V04 LABEL RECORD STANDARD BLOCK 8500. 01 RV04 PIC X(2).		VIV00900
FD V05 LABEL RECORD STANDARD BLOCK 8500. 01 RV05 PIC X(2).		VIV00910
FD V06A LABEL RECORD STANDARD BLOCK 8500. 01 RV06A PIC X(2).		VIV00920
FD V06B LABEL RECORD STANDARD BLOCK 8500. 01 RV06B PIC X(2).		VIV00930
FD V06C LABEL RECORD STANDARD BLOCK 8500. 01 RV06C PIC X(2).		VIV00940
FD V06D LABEL RECORD STANDARD BLOCK 8500. 01 RV06D PIC X(2).		VIV00950
FD V06E LABEL RECORD STANDARD BLOCK 8500. 01 RV06E PIC X(2).		VIV00960
FD V06F LABEL RECORD STANDARD BLOCK 8500. 01 RV06F PIC X(2).		VIV00970
FD V07 LABEL RECORD STANDARD BLOCK 19000. 01 RV07 PIC X(1).		VIV00980
FD V08A LABEL RECORD STANDARD BLOCK 19000. 01 RVC8A PIC X(1).		VIV00990
FD V08B LABEL RECORD STANDARD BLOCK 19000. 01 RV08B PIC X(1).		VIV01000
FD V09 LABEL RECORD STANDARD BLOCK 19000. 01 RV09 PIC X(1).		VIV01010
FD V10 LABEL RECORD STANDARD BLOCK 19000. 01 RV10 PIC X(1).		VIV01020
FD V11 LABEL RECORD STANDARD BLOCK 19000. 01 RV11 PIC X(1).		VIV01030

FD	V12	LABEL RECORD STANDARD BLOCK 19000.	01	RV12	PIC X(1).	VIV01040
FD	V13A	LABEL RECORD STANDARD BLOCK 19000.	01	RV13A	PIC X(1).	VIV01050
FD	V13B	LABEL RECORD STANDARD BLOCK 19000.	01	KV13B	PIC X(1).	VIV01060
FD	V13C	LABEL RECORD STANDARD BLOCK 19000.	01	EV13C	PIC X(1).	VIV01070
FD	V13D	LABEL RECORD STANDARD BLOCK 19000.	01	EV13D	PIC X(1).	VIV01080
FD	V13E	LABEL RECORD STANDARD BLOCK 19000.	01	KV13E	PIC X(1).	VIV01090
FD	V13F	LABEL RECORD STANDARD BLOCK 19000.	01	EV13F	PIC X(1).	VIV01100
FD	V13G	LABEL RECORD STANDARD BLOCK 19000.	01	EV13G	PIC X(1).	VIV01110
FD	V13H	LABEL RECORD STANDARD BLOCK 19000.	01	EV13H	PIC X(1).	VIV01120
FD	V13I	LABEL RECORD STANDARD BLOCK 19000.	01	EV13I	PIC X(1).	VIV01130
FD	V13J	LABEL RECORD STANDARD BLOCK 19000.	01	RV13J	PIC X(1).	VIV01140
FD	V13K	LABEL RECORD STANDARD BLOCK 19000.	01	RV13K	PIC X(1).	VIV01150
FD	V13L	LABEL RECORD STANDARD BLOCK 19000.	01	RV13L	PIC X(1).	VIV01160
FD	FINCA	LABEL RECORD STANDARD BLOCK 19000.	01	RFINCA	PIC X(1).	VIV01170
FD	GANADO	LABEL RECORD STANDARD BLOCK 19000.	01	RGANADO	PIC X.	VIV01180
FD	TOTALP	LABEL RECORD STANDARD BLOCK 19000.	01	RTOTALP	PIC XX.	VIV01190
FD	TOTALH	LABEL RECORD STANDARD BLOCK 19000.	01	RTOTALH	PIC XX.	VIV01200
FD	TOTALM	LABEL RECORD STANDARD BLOCK 19000.	01	RTOTALM	PIC XX.	VIV01210
*						VIV01220
		WORKING-STORAGE SECTION.				VIV01230
77	TOTAL-CENSO	PIC	9(8)	COMP VALUE ZERO.		VIV01240
77	TOTAL-HOGARES	PIC	9(8)	COMP VALUE ZERO.		VIV01250
*						VIV01260
*		EL SGTE REGISTRO DEBE MODIFICARSE PARA EL CENSO PARTICULAR				VIV01270
*		TANTO LA IDENTIFICACION COMO LAS VARIABLES SUSTANTIVAS				VIV01280
*						VIV01290
01	REGISTRO-CENSO.					VIV01300
03	IDENTIFICACION.					VIV01310
05	SEGMENTO.					VIV01320
07	PROVINCIA	PIC X.				VIV01330
07	CANTON	PIC XX.				VIV01340
07	DISTRITO	PIC XX.				VIV01350
07	FILLER	PIC XXX.				VIV01360
05	EVIV	PIC XXX.				VIV01370
05	EHOG	PIC 9.				VIV01380
03	TIPO-REG	PIC 9.				VIV01390
03	REGISTRO-VIV.					VIV01400
05	FILLER	PIC XX.				VIV01410
05	EV01	PIC X(1).				VIV01420
05	EV02A	PIC X(1).				VIV01430
05	EV02B	PIC X(5).				VIV01440
05	EV03	PIC X(2).				VIV01450
05	EV04	PIC X(2).				VIV01460
05	EV05	PIC X(2).				VIV01470
05	EV06A	PIC X(2).				VIV01480
05	EV06B	PIC X(2).				VIV01490
05	EV06C	PIC X(2).				VIV01500
05	EV06D	PIC X(2).				VIV01510
05	EV06E	PIC X(2).				VIV01520
05	EV06F	PIC X(2).				VIV01530
05	EV07	PIC X(1).				VIV01540
05	EV08A	PIC X(1).				VIV01550
05	EV08B	PIC X(1).				VIV01560

05	EV09	PIC X(1).	VIV01570
05	EV10	PIC X(1).	VIV01580
05	EV11	PIC X(1).	VIV01590
05	EV12	PIC X(1).	VIV01600
05	EV13A	PIC X(1).	VIV01610
05	EV13B	PIC X(1).	VIV01620
05	EV13C	PIC X(1).	VIV01630
05	EV13D	PIC X(1).	VIV01640
05	EV13E	PIC X(1).	VIV01650
05	EV13F	PIC X(1).	VIV01660
05	EV13G	PIC X(1).	VIV01670
05	EV13H	PIC X(1).	VIV01680
05	EV13I	PIC X(1).	VIV01690
05	EV13J	PIC X(1).	VIV01700
05	EV13K	PIC X(1).	VIV01710
05	EV13L	PIC X(1).	VIV01720
05	EFINCA	PIC X(1).	VIV01730
05	EGANADO	PIC X.	VIV01740
05	ETOTALP	PIC XX.	VIV01750
05	ETOTALH	PIC XX.	VIV01760
05	ETOTALM	PIC XX.	VIV01770
05	FILLER	PIC X(4).	VIV01780
03	REGISTRO-POB	REDEFINES REGISTRO-VIV.	VIV01790
05	FILLER	PIC X(2).	VIV01800
05	EP01	PIC X(1).	VIV01810
05	EP02	PIC 9.	VIV01820
05	FILLER	PIC X(53).	VIV01830
*			VIV01840
01	VARIABLES-A-GRABAR.		VIV01850
03	SVIV	PIC X(3).	VIV01860
03	SHOG	PIC X(1).	VIV01870
03	SV01	PIC X(1).	VIV01880
03	SV02A	PIC X(1).	VIV01890
03	SV02B	PIC X(5).	VIV01900
03	SV03	PIC X(2).	VIV01910
03	SV04	PIC X(2).	VIV01920
03	SV05	PIC X(2).	VIV01930
03	SV06A	PIC X(2).	VIV01940
03	SV06B	PIC X(2).	VIV01950
03	SV06C	PIC X(2).	VIV01960
03	SV06D	PIC X(2).	VIV01970
03	SV06E	PIC X(2).	VIV01980
03	SV06F	PIC X(2).	VIV01990
03	SV07	PIC X(1).	VIV02000
03	SV08A	PIC X(1).	VIV02010
03	SV08B	PIC X(1).	VIV02020
03	SV09	PIC X(1).	VIV02030
03	SV10	PIC X(1).	VIV02040
03	SV11	PIC X(1).	VIV02050
03	SV12	PIC X(1).	VIV02060
03	SV13A	PIC X(1).	VIV02070
03	SV13B	PIC X(1).	VIV02080
03	SV13C	PIC X(1).	VIV02090

V10	VIV02630
V11	VIV02640
V12	VIV02650
V13A	VIV02660
V13B	VIV02670
V13C	VIV02680
V13D	VIV02690
V13E	VIV02700
V13F	VIV02710
V13G	VIV02720
V13H	VIV02730
V13I	VIV02740
V13J	VIV02750
V13K	VIV02760
V13L	VIV02770
FINCA	VIV02780
GANADO	VIV02790
TOTALP	VIV02800
TOTALH	VIV02810
TOTALM.	VIV02820
READ CENSO INTO REGISTRO-CENSO AT END MOVE 1 TO FIN-ARCH.	VIV02830
MOVE ZERO TO SEXO (1), SEXO (2).	VIV02840
PERFORM PROCESA-HOGAR.	VIV02850
PERFORM CICLO THRU CICLO-FIN UNTIL FIN-ARCHIVO.	VIV02860
DISPLAY " TOTAL REGISTROS LEIDOS ", TOTAL-CENSO.	VIV02870
DISPLAY " TOTAL HOGARES GRABADOS ", TOTAL-HOGARES.	VIV02880
CLOSE CENSO	VIV02890
VIV	VIV02900
HOG	VIV02910
V01	VIV02920
V02A	VIV02930
V02B	VIV02940
V03	VIV02950
V04	VIV02960
V05	VIV02970
V06A	VIV02980
V06B	VIV02990
V06C	VIV03000
V06D	VIV03010
V06E	VIV03020
V06F	VIV03030
V07	VIV03040
V08A	VIV03050
V08B	VIV03060
V09	VIV03070
V10	VIV03080
V11	VIV03090
V12	VIV03100
V13A	VIV03110
V13B	VIV03120
V13C	VIV03130
V13D	VIV03140
V13E	VIV03150

```

V13F VIV03160
V13G VIV03170
V13H VIV03180
V13I VIV03190
V13J VIV03200
V13K VIV03210
V13L VIV03220
FINCA VIV03230
GANADO VIV03240
TOTALP VIV03250
TOTALH VIV03260
TOTALM. VIV03270
STOP RUN. VIV03280
* VIV03290
CICLO. VIV03300
  READ CENSO INTO REGISTRO-CENSO AT END MOVE 1 TO FIN-ARCH VIV03310
  MOVE 1 TO TIPO-REG. VIV03320
  ADD 1 TO TOTAL-CENSO. VIV03330
  EXAMINE REGISTRO-CENSO REPLACING ALL SPACES BY ZERO. VIV03340
* ESTE EXAMINE ES NECESARIO PUES EL SOFTWARE DE TRANSMISION VIV03350
* CORTA LOS BLANCOS DEL FINAL DE CADA BLOQUE VIV03360
  IF TIPO-REG = 2 VIV03370
    ADD 1 TO SEXO (EP02) VIV03380
  ELSE VIV03390
    PERFORM PROCESA-TOTAL VIV03400
    IF NOT FIN-ARCHIVO PERFORM PROCESA-HOGAR. VIV03410
CICLO-FIN. VIV03420
  EXIT. VIV03430
PROCESA-HOGAR. VIV03440
  MOVE EVIV TO SVIV. WRITE RVIV FROM SVIV. VIV03450
  MOVE EHOg TO SHOG. WRITE RHOG FROM SHOG. VIV03460
  MOVE EFINCA TO SFINCA. WRITE RFINCA FROM SFINCA. VIV03470
  MOVE EGANADO TO SGANADO. WRITE RGANADO FROM SGANADO. VIV03480
  IF EHOg = 1 VIV03490
    PERFORM PROCESA-HOGAR1. VIV03500
* LAS VARIABLES DEL PRIMER HOGAR SE GUARDAN EN ESTA RUTINA VIV03510
* PARA SER PASADAS A LOS SIGUIENTES HOGARES DE LA VIVIENDA. VIV03520
* ESTO ES NECESARIO PUES ALGUNOS DATOS DE LA VIVIENDA NO SE VIV03530
* DIGITARON PARA EL SEGUNDO HOGAR EN ADELANTE, Y PARA UNA VIV03540
* UTILIZACION EFECTIVA DE ESTOS DATOS A TRAVES DE REDATAM ES VIV03550
* ES NECESARIO TENERLOS EN TODOS LOS REGISTROS DE HOGAR VIV03560
* VIV03570
  WRITE RV01 FROM SV01. VIV03580
  WRITE RV02A FROM SV02A. VIV03590
  WRITE RV02B FROM SV02B. VIV03600
  WRITE RV03 FROM SV03. VIV03610
  WRITE RV04 FROM SV04. VIV03620
  WRITE RV05 FROM SV05. VIV03630
  WRITE RV06A FROM SV06A. VIV03640
  WRITE RV06B FROM SV06B. VIV03650
  WRITE RV06C FROM SV06C. VIV03660
  WRITE RV06D FROM SV06D. VIV03670
  WRITE RV06E FROM SV06E. VIV03680

```

```

WRITE RV06F      FROM SV06F.      VIV03690
WRITE RV07       FROM SV07.       VIV03700
WRITE RV08A     FROM SV08A.      VIV03710
WRITE RV08B     FROM SV08B.      VIV03720
WRITE RV09      FROM SV09.       VIV03730
WRITE RV10     FROM SV10.       VIV03740
WRITE RV11     FROM SV11.       VIV03750
WRITE RV12     FROM SV12.       VIV03760
WRITE RV13A    FROM SV13A.      VIV03770
WRITE RV13B    FROM SV13B.      VIV03780
WRITE RV13C    FROM SV13C.      VIV03790
WRITE RV13D    FROM SV13D.      VIV03800
WRITE RV13E    FROM SV13E.      VIV03810
WRITE RV13F    FROM SV13F.      VIV03820
WRITE RV13G    FROM SV13G.      VIV03830
WRITE RV13H    FROM SV13H.      VIV03840
WRITE RV13I    FROM SV13I.      VIV03850
WRITE RV13J    FROM SV13J.      VIV03860
WRITE RV13K    FROM SV13K.      VIV03870
WRITE RV13L    FROM SV13L.      VIV03880
*
ADD 1 TO TOTAL-HOGARES.          VIV03890
PROCESA-HOGAR1.                  VIV03900
* EN ESTE PARRAFO SE HAN HECHO ALGUNAS CONSISTENCIAS DE VARIABLES VIV03910
* QUE APARECIERON CON DATOS INVALIDOS EN LAS DISTRIBUCIONES DE VIV03920
* DE FRECUENCIA EFECTUADAS      VIV03930
*                                VIV03940
*                                VIV03950
MOVE EV01      TO SV01.          VIV03960
MOVE EV02A    TO SV02A.         VIV03970
IF EV02B NOT NUMERIC MOVE ALL "0" TO SV02B VIV03980
ELSE MOVE EV02B TO SV02B.      VIV03990
IF EV03 NOT NUMERIC MOVE ALL "0" TO SV03 VIV04000
ELSE MOVE EV03 TO SV03.       VIV04010
MOVE EV04     TO SV04.          VIV04020
MOVE EV05     TO SV05.          VIV04030
MOVE EV06A    TO SV06A.         VIV04040
IF EV06B NOT NUMERIC MOVE "00" TO SV06B VIV04050
ELSE MOVE EV06B TO SV06B.     VIV04060
MOVE EV06C    TO SV06C.         VIV04070
MOVE EV06D    TO SV06D.         VIV04080
MOVE EV06E    TO SV06E.         VIV04090
MOVE EV06F    TO SV06F.         VIV04100
MOVE EV07     TO SV07.          VIV04110
MOVE EV08A    TO SV08A.         VIV04120
MOVE EV08B    TO SV08B.         VIV04130
MOVE EV09     TO SV09.          VIV04140
MOVE EV10     TO SV10.          VIV04150
MOVE EV11     TO SV11.          VIV04160
MOVE EV12     TO SV12.          VIV04170
MOVE EV13A    TO SV13A.         VIV04180
MOVE EV13B    TO SV13B.         VIV04190
MOVE EV13C    TO SV13C.         VIV04200
MOVE EV13D    TO SV13D.         VIV04210

```

```

                MOVE EV13E      TO SV13E      VIV04220
                MOVE EV13F      TO SV13F      VIV04230
                MOVE EV13G      TO SV13G      VIV04240
                MOVE EV13H      TO SV13H      VIV04250
                MOVE EV13I      TO SV13I      VIV04260
                MOVE EV13J      TO SV13J      VIV04270
                MOVE EV13K      TO SV13K      VIV04280
                MOVE EV13L      TO SV13L      VIV04290
*
  PROCESA-TOTAL.                                VIV04300
* EN ESTE PARRAFO SE CONTABILIZA EL TOTAL DE HOMBRES Y MUJERES VIV04310
* DE CADA HOGAR, ASI COMO EL TOTAL DE POBLACION. FUE NECESARIO VIV04320
* PUES SE DESEABA TENER ESTOS DATOS EN EL REGISTRO DE VIVIENDA VIV04330
* DE REDATAM VIV04340
                MOVE SEXO (1)  TO STOTALH.  WRITE RTOTALH FROM STOTALH. VIV04350
                MOVE SEXO (2)  TO STOTALM.  WRITE RTOTALM FROM STOTALM. VIV04360
                ADD STOTALH, STOTALM GIVING STOTALP. VIV04370
                WRITE RTOTALP FROM STOTALP. VIV04380
*
                MOVE ZERO TO SEXO (1) SEXO (2). VIV04390
  FIN-PROGRAMA. VIV04400
  EXIT. VIV04410
//GO.SYSUDUMP DD SYSOUT=A VIV04420
//SYSOUT DD SYSOUT=A VIV04430
//CENSO DD UNIT=TAPE, VIV04440
// VOL=SER=(R402,R403,R404,R405,R406),DSN=CR84CENS.ORIGIN80 VIV04450
//VIV DD UNIT=3350,VOL=SER=OSWORK,DSN=CR84CENS.VIV, VIV04460
// DCB=(RECFM=FB,LRECL=3,BLKSIZE=18000),DISP=(OLD), VIV04470
// SPACE=(TRK,(90,30),RLSE) VIV04480
//HOG DD UNIT=3350,VOL=SER=OSWORK,DSN=CR84CENS.HOG, VIV04490
// DCB=(RECFM=FB,LRECL=1,BLKSIZE=18000),DISP=(OLD), VIV04500
// SPACE=(TRK,(30,10),RLSE) VIV04510
//VO1 DD UNIT=3350,VOL=SER=OSWORK,DSN=CR84CENS.VO1, VIV04520
// DCB=(RECFM=FB,LRECL=1,BLKSIZE=18000),DISP=(OLD), VIV04530
// SPACE=(TRK,(30,20),RLSE) VIV04540
//VO2A DD UNIT=3350,VOL=SER=OSWORK,DSN=CR84CENS.VO2A, VIV04550
// DCB=(RECFM=FB,LRECL=1,BLKSIZE=18000),DISP=(OLD), VIV04560
// SPACE=(TRK,(30,20),RLSE) VIV04570
//VO2B DD UNIT=3350,VOL=SER=OSWORK,DSN=CR84CENS.VO2B, VIV04580
// DCB=(RECFM=FB,LRECL=5,BLKSIZE=18000),DISP=(OLD), VIV04590
// SPACE=(TRK,(150,20),RLSE) VIV04600
//VO3 DD UNIT=3350,VOL=SER=OSWORK,DSN=CR84CENS.VO3, VIV04610
// DCB=(RECFM=FB,LRECL=2,BLKSIZE=18000),DISP=(OLD), VIV04620
// SPACE=(TRK,(60,20),RLSE) VIV04630
//VO4 DD UNIT=3350,VOL=SER=OSWORK,DSN=CR84CENS.VO4, VIV04640
// DCB=(RECFM=FB,LRECL=2,BLKSIZE=18000),DISP=(OLD), VIV04650
// SPACE=(TRK,(60,10),RLSE) VIV04660
//VO5 DD UNIT=3350,VOL=SER=OSWORK,DSN=CR84CENS.VO5, VIV04670
// DCB=(RECFM=FB,LRECL=2,BLKSIZE=18000),DISP=(OLD), VIV04680
// SPACE=(TRK,(60,20),RLSE) VIV04690
//VO6A DD UNIT=3350,VOL=SER=OSWORK,DSN=CR84CENS.VO6A, VIV0700
// DCB=(RECFM=FB,LRECL=2,BLKSIZE=18000),DISP=(OLD), VIV0710
// SPACE=(TRK,(60,20),RLSE) VIV0720
//VO6B DD UNIT=3350,VOL=SER=OSWORK,DSN=CR84CENS.VO6B, VIV0730
// DCB=(RECFM=FB,LRECL=2,BLKSIZE=18000),DISP=(OLD), VIV0740
// SPACE=(TRK,(60,20),RLSE) VIV0740

```

```
//V06B DD UNIT=3350,VOL=SER=OSWORK,DSN=CR84CENS.V06B, VIV04750
// DCB=(RECFM=FB,LRECL=2,BLKSIZE=18000),DISP=(OLD), VIV04760
// SPACE=(TRK,(60,10),RLSE) VIV04770
//V06C DD UNIT=3350,VOL=SER=OSWORK,DSN=CR84CENS.V06C, VIV04780
// DCB=(RECFM=FB,LRECL=2,BLKSIZE=18000),DISP=(OLD), VIV04790
// SPACE=(TRK,(60,10),RLSE) VIV04800
//V06D DD UNIT=3350,VOL=SER=OSWORK,DSN=CR84CENS.V06D, VIV04810
// DCB=(RECFM=FB,LRECL=2,BLKSIZE=18000),DISP=(OLD), VIV04820
// SPACE=(TRK,(60,10),RLSE) VIV04830
//V06E DD UNIT=3350,VOL=SER=OSWORK,DSN=CR84CENS.V06E, VIV04840
// DCB=(RECFM=FB,LRECL=2,BLKSIZE=18000),DISP=(OLD), VIV04850
// SPACE=(TRK,(60,10),RLSE) VIV04860
//V06F DD UNIT=3350,VOL=SER=OSWORK,DSN=CR84CENS.V06F, VIV04870
// DCB=(RECFM=FB,LRECL=2,BLKSIZE=18000),DISP=(OLD), VIV04880
// SPACE=(TRK,(60,10),RLSE) VIV04890
//V07 DD UNIT=3350,VOL=SER=OSWORK,DSN=CR84CENS.V07, VIV04900
// DCB=(RECFM=FB,LRECL=1,BLKSIZE=18000),DISP=(OLD), VIV04910
// SPACE=(TRK,(30,10),RLSE) VIV04920
//V08A DD UNIT=3350,VOL=SER=OSWORK,DSN=CR84CENS.V08A, VIV04930
// DCB=(RECFM=FB,LRECL=1,BLKSIZE=18000),DISP=(OLD), VIV04940
// SPACE=(TRK,(30,10),RLSE) VIV04950
//V08B DD UNIT=3350,VOL=SER=OSWORK,DSN=CR84CENS.V08B, VIV04960
// DCB=(RECFM=FB,LRECL=1,BLKSIZE=18000),DISP=(OLD), VIV04970
// SPACE=(TRK,(30,10),RLSE) VIV04980
//V09 DD UNIT=3350,VOL=SER=OSWORK,DSN=CR84CENS.V09, VIV04990
// DCB=(RECFM=FB,LRECL=1,BLKSIZE=18000),DISP=(OLD), VIV05000
// SPACE=(TRK,(30,10),RLSE) VIV05010
//V10 DD UNIT=3350,VOL=SER=OSWORK,DSN=CR84CENS.V10, VIV05020
// DCB=(RECFM=FB,LRECL=1,BLKSIZE=18000),DISP=(OLD), VIV05030
// SPACE=(TRK,(30,10),RLSE) VIV05040
//V11 DD UNIT=3350,VOL=SER=OSWORK,DSN=CR84CENS.V11, VIV05050
// DCB=(RECFM=FB,LRECL=1,BLKSIZE=18000),DISP=(OLD), VIV05060
// SPACE=(TRK,(30,10),RLSE) VIV05070
//V12 DD UNIT=3350,VOL=SER=OSWORK,DSN=CR84CENS.V12, VIV05080
// DCB=(RECFM=FB,LRECL=1,BLKSIZE=18000),DISP=(OLD), VIV05090
// SPACE=(TRK,(30,10),RLSE) VIV05100
//V13A DD UNIT=3350,VOL=SER=OSWORK,DSN=CR84CENS.V13A, VIV05110
// DCB=(RECFM=FB,LRECL=1,BLKSIZE=18000),DISP=(OLD), VIV05120
// SPACE=(TRK,(30,10),RLSE) VIV05130
//V13B DD UNIT=3350,VOL=SER=OSWORK,DSN=CR84CENS.V13B, VIV05140
// DCB=(RECFM=FB,LRECL=1,BLKSIZE=18000),DISP=(OLD), VIV05150
// SPACE=(TRK,(30,10),RLSE) VIV05160
//V13C DD UNIT=3350,VOL=SER=OSWORK,DSN=CR84CENS.V13C, VIV05170
// DCB=(RECFM=FB,LRECL=1,BLKSIZE=18000),DISP=(OLD), VIV05180
// SPACE=(TRK,(30,10),RLSE) VIV05190
//V13D DD UNIT=3350,VOL=SER=OSWORK,DSN=CR84CENS.V13D, VIV05200
// DCB=(RECFM=FB,LRECL=1,BLKSIZE=18000),DISP=(OLD), VIV05210
// SPACE=(TRK,(30,10),RLSE) VIV05220
//V13E DD UNIT=3350,VOL=SER=OSWORK,DSN=CR84CENS.V13E, VIV05230
// DCB=(RECFM=FB,LRECL=1,BLKSIZE=18000),DISP=(OLD), VIV05240
// SPACE=(TRK,(30,10),RLSE) VIV05250
//V13F DD UNIT=3350,VOL=SER=OSWORK,DSN=CR84CENS.V13F, VIV05260
// DCB=(RECFM=FB,LRECL=1,BLKSIZE=18000),DISP=(OLD), VIV05270
```

```
// SPACE=(TRK,(30,10),RLSE) VIV05280
//V13G DD UNIT=3350,VOL=SER=OSWORK,DSN=CR84CENS.V13G, VIV05290
// DCB=(RECFM=FB,LRECL=1,BLKSIZE=18000),DISP=(OLD), VIV05300
// SPACE=(TRK,(30,10),RLSE) VIV05310
//V13H DD UNIT=3350,VOL=SER=OSWORK,DSN=CR84CENS.V13H, VIV05320
// DCB=(RECFM=FB,LRECL=1,BLKSIZE=18000),DISP=(OLD), VIV05330
// SPACE=(TRK,(30,10),RLSE) VIV05340
//V13I DD UNIT=3350,VOL=SER=OSWORK,DSN=CR84CENS.V13I, VIV05350
// DCB=(RECFM=FB,LRECL=1,BLKSIZE=18000),DISP=(OLD), VIV05360
// SPACE=(TRK,(30,10),RLSE) VIV05370
//V13J DD UNIT=3350,VOL=SER=OSWORK,DSN=CR84CENS.V13J, VIV05380
// DCB=(RECFM=FB,LRECL=1,BLKSIZE=18000),DISP=(OLD), VIV05390
// SPACE=(TRK,(30,10),RLSE) VIV05400
//V13K DD UNIT=3350,VOL=SER=OSWORK,DSN=CR84CENS.V13K, VIV05410
// DCB=(RECFM=FB,LRECL=1,BLKSIZE=18000),DISP=(OLD), VIV05420
// SPACE=(TRK,(30,10),RLSE) VIV05430
//V13L DD UNIT=3350,VOL=SER=OSWORK,DSN=CR84CENS.V13L, VIV05440
// DCB=(RECFM=FB,LRECL=1,BLKSIZE=18000),DISP=(OLD), VIV05450
// SPACE=(TRK,(30,10),RLSE) VIV05460
//FINCA DD UNIT=3350,VOL=SER=OSWORK,DSN=CR84CENS.FINCA, VIV05470
// DCB=(RECFM=FB,LRECL=1,BLKSIZE=18000),DISP=(OLD), VIV05480
// SPACE=(TRK,(30,10),RLSE) VIV05490
//GANADO DD UNIT=3350,VOL=SER=OSWORK,DSN=CR84CENS.GANADO, VIV05500
// DCB=(RECFM=FB,LRECL=1,BLKSIZE=18000),DISP=(OLD), VIV05510
// SPACE=(TRK,(30,10),RLSE) VIV05520
//TOTALP DD UNIT=3350,VOL=SER=OSWORK,DSN=CR84CENS.TOTALP, VIV05530
// DCB=(RECFM=FB,LRECL=2,BLKSIZE=18000),DISP=(OLD), VIV05540
// SPACE=(TRK,(60,10),RLSE) VIV05550
//TOTALH DD UNIT=3350,VOL=SER=OSWORK,DSN=CR84CENS.TOTALH, VIV05560
// DCB=(RECFM=FB,LRECL=2,BLKSIZE=18000),DISP=(OLD), VIV05570
// SPACE=(TRK,(60,10),RLSE) VIV05580
//TOTALM DD UNIT=3350,VOL=SER=OSWORK,DSN=CR84CENS.TOTALM, VIV05590
// DCB=(RECFM=FB,LRECL=2,BLKSIZE=18000),DISP=(OLD), VIV05600
// SPACE=(TRK,(60,10),RLSE) VIV05610
```

 APENDICE F.3 - PROGRAMA POBLACION

```
//STEP1 EXEC COBUCLG                                POB00010
  IDENTIFICATION DIVISION.                            POB00020
  PROGRAM-ID. POBLACION.                              POB00030
  AUTHOR. HARRY HERNANDEZ.                            POB00040
  DATE-WRITTEN. 15 MAYO 1987.                         POB00050
  DATE-COMPILED.                                     POB00060
  REMARKS. ----- POB00070
    ! ESTE PROGRAMA PERTENECE A LA INSTALACION DEL SISTEMA ! POB00080
    ! REDATAM PARA EL CENSO DE COSTA RICA DE 1984.        ! POB00090
    ! SU FUNCION ES DIVIDIR LOS REGISTROS DE POBLACION    ! POB00100
    ! DE DICHO CENSO Y SEPARAR CADA VARIABLE EN UN ARCHIVO ! POB00110
    ! DIFERENTE.                                          ! POB00120
    ! ENTRADA:                                            ! POB00130
    ! -CINTAS DEL CENSO DE POBLACION Y VIVIENDA          ! POB00140
    ! PROCESO:                                           ! POB00150
    ! -SELECCIONA LOS REGISTROS DE POBLACION             ! POB00160
    ! -SEPARA CADA REGISTRO EN LAS VARIABLES A GENERAR   ! POB00170
    ! PARA REDATAM                                       ! POB00180
    ! -CREA ALGUNAS VARIABLES NUEVAS                     ! POB00190
    ! -GRABA LAS VARIABLES                               ! POB00200
    ! SALIDA:                                            ! POB00210
    ! -TANTOS ARCHIVOS COMO VARIABLES SE GENEREN PARA   ! POB00220
    ! EL SISTEMA READATAM                                ! POB00230
    !                                                    ! POB00240
    !                                                    ! POB00250
    ----- POB00260
  ENVIRONMENT DIVISION.                               POB00270
  CONFIGURATION SECTION.                             POB00280
  SOURCE-COMPUTER. IBM-370.                           POB00290
  OBJECT-COMPUTER. IBM-370.                           POB00300
  INPUT-OUTPUT SECTION.                              POB00310
  FILE-CONTROL.                                       POB00320
    SELECT CENSO ASSIGN TO UT-3410-S-CENSO.            POB00330
* ---SIGUE LA DEFINICION DE UN ARCHIVO PARA C/VARIABLE DE POB. POB00340
    SELECT P01 ASSIGN TO DA-3350-S-P01.               POB00350
    SELECT P02 ASSIGN TO DA-3350-S-P02.               POB00360
    SELECT P03 ASSIGN TO DA-3350-S-P03.               POB00370
    SELECT NAQUI ASSIGN TO DA-3350-S-NAQUI.           POB00380
    SELECT P04A ASSIGN TO DA-3350-S-P04A.             POB00390
    SELECT P04B ASSIGN TO DA-3350-S-P04B.             POB00400
    SELECT P04C ASSIGN TO DA-3350-S-P04C.             POB00410
    SELECT P04D ASSIGN TO DA-3350-S-P04D.             POB00420
    SELECT P05 ASSIGN TO DA-3350-S-P05.               POB00430
    SELECT P06 ASSIGN TO DA-3350-S-P06.               POB00440
    SELECT RAQUI ASSIGN TO DA-3350-S-RAQUI.           POB00450
    SELECT P07A ASSIGN TO DA-3350-S-P07A.             POB00460
    SELECT P07B ASSIGN TO DA-3350-S-P07B.             POB00470
    SELECT P08 ASSIGN TO DA-3350-S-P08.               POB00480
    SELECT P09 ASSIGN TO DA-3350-S-P09.               POB00490
    SELECT P10 ASSIGN TO DA-3350-S-P10.               POB00500
```

```

SELECT P11          ASSIGN TO DA-3350-S-P11.          POB00510
SELECT P12          ASSIGN TO DA-3350-S-P12.          POB00520
SELECT P13          ASSIGN TO DA-3350-S-P13.          POB00530
SELECT P14          ASSIGN TO DA-3350-S-P14.          POB00540
SELECT P15          ASSIGN TO DA-3350-S-P15.          POB00550
SELECT P16          ASSIGN TO DA-3350-S-P16.          POB00560
SELECT P17          ASSIGN TO DA-3350-S-P17.          POB00570
SELECT P18          ASSIGN TO DA-3350-S-P18.          POB00580
*
DATA DIVISION.          POB00590
FILE SECTION.          POB00600
FD CENSO              POB00610
  BLOCK CONTAINS 0 RECORDS          POB00620
  LABEL RECORDS ARE STANDARD        POB00630
  RECORD CONTAINS 70 CHARACTERS.    POB00640
*
* ----- DEBE MODIFICARSE AL LARGO DEL REGISTRO DEL CENSO PARTIC. POB00650
*
01 REG-CENSO          PIC X(70).      POB00660
*
* ----SIGUE LA DEFINICION DE LOS ARCHIVOS PARA CADA VAR. POBLACION POB00670
*
FD P01 LABEL RECORD STANDARD BLOCK 0. 01 RP01 PIC X(1).      POB00680
FD P02 LABEL RECORD STANDARD BLOCK 0. 01 RP02 PIC X(1).      POB00690
FD P03 LABEL RECORD STANDARD BLOCK 0. 01 RP03 PIC X(2).      POB00700
FD NAQUI LABEL RECORD STANDARD BLOCK 0. 01 RNAQUI PIC X(1).   POB00710
FD P04A LABEL RECORD STANDARD BLOCK 0. 01 RP04A PIC X(1).     POB00720
FD P04B LABEL RECORD STANDARD BLOCK 0. 01 RP04B PIC X(2).     POB00730
FD P04C LABEL RECORD STANDARD BLOCK 0. 01 RP04C PIC X(3).     POB00740
FD P04D LABEL RECORD STANDARD BLOCK 0. 01 RP04D PIC X(2).     POB00750
FD P05 LABEL RECORD STANDARD BLOCK 0. 01 RP05 PIC X(3).      POB00760
FD P06 LABEL RECORD STANDARD BLOCK 0. 01 RP06 PIC X(1).      POB00770
FD RAQUI LABEL RECORD STANDARD BLOCK 0. 01 RRAQUI PIC X(1).   POB00780
FD P07A LABEL RECORD STANDARD BLOCK 0. 01 RP07A PIC X(1).     POB00790
FD P07B LABEL RECORD STANDARD BLOCK 0. 01 RP07B PIC X(2).     POB00800
FD P08 LABEL RECORD STANDARD BLOCK 0. 01 RP08 PIC X(1).      POB00810
FD P09 LABEL RECORD STANDARD BLOCK 0. 01 RP09 PIC X(2).      POB00820
FD P10 LABEL RECORD STANDARD BLOCK 0. 01 RP10 PIC X(1).      POB00830
FD P11 LABEL RECORD STANDARD BLOCK 0. 01 RP11 PIC X(1).      POB00840
FD P12 LABEL RECORD STANDARD BLOCK 0. 01 RP12 PIC X(1).      POB00850
FD P13 LABEL RECORD STANDARD BLOCK 0. 01 RP13 PIC X(3).      POB00860
FD P14 LABEL RECORD STANDARD BLOCK 0. 01 RP14 PIC X(1).      POB00870
FD P15 LABEL RECORD STANDARD BLOCK 0. 01 RP15 PIC X(4).      POB00880
FD P16 LABEL RECORD STANDARD BLOCK 0. 01 RP16 PIC X(2).      POB00890
FD P17 LABEL RECORD STANDARD BLOCK 0. 01 RP17 PIC X(2).      POB00900
FD P18 LABEL RECORD STANDARD BLOCK 0. 01 RP18 PIC X(2).      POB00910
*
*
WORKING-STORAGE SECTION. POB00920
77 TOTAL-CENSO          PIC 9(8) COMP VALUE ZERO.          POB00930
77 TOTAL-PERSONAS      PIC 9(8) COMP VALUE ZERO.          POB00940
*
* EL SGTE REGISTRO DEBE MODIFICARSE PARA EL CENSO PARTICULAR POB00950
  
```

* TANTO LA IDENTIFICACION COMO LAS VARIABLES SUSTANTIVAS	POB01040
*	POB01050
01 REGISTRO-CENSO.	POB01060
03 IDENTIFICACION.	POB01070
05 SEGMENTO.	POB01080
07 PROVINCIA PIC X.	POB01090
07 CANTON PIC XX.	POB01100
07 DISTRITO PIC 99.	POB01110
07 FILLER PIC XXX.	POB01120
05 EVIV PIC XXX.	POB01130
05 EHOZ PIC 9.	POB01140
03 TIPO-REG PIC 9.	POB01150
03 REGISTRO-VIV.	POB01160
05 FILLER PIC X(57).	POB01170
03 REGISTRO-POB REDEFINES REGISTRO-VIV.	POB01180
05 FILLER PIC XX.	POB01190
05 EP01 PIC X(1).	POB01200
05 EP02 PIC X(1).	POB01210
05 EP03 PIC X(2).	POB01220
05 EP04.	POB01230
07 EP04A PIC X(1).	POB01240
07 EP04B PIC X(2).	POB01250
07 EP04C PIC 9(2).	POB01260
05 EP04R REDEFINES EP04.	POB01270
07 EP04X PIC X(2).	POB01280
07 EP04Y PIC 9(3).	POB01290
05 EP04D PIC 9(4).	POB01300
05 EP04DR REDEFINES EP04D.	POB01310
07 FILLER PIC X(2).	POB01320
07 EP04DX PIC 9(2).	POB01330
05 EP05 PIC X(3).	POB01340
05 EP06 PIC X(1).	POB01350
05 EP07.	POB01360
07 EP07A PIC X(1).	POB01370
07 EP07B PIC X(2).	POB01380
05 EP08 PIC X(1).	POB01390
05 EP09 PIC X(2).	POB01400
05 EP10 PIC X(1).	POB01410
05 EP11 PIC X(1).	POB01420
05 EP12 PIC X(1).	POB01430
05 EP13 PIC X(3).	POB01440
05 EP14 PIC X(1).	POB01450
05 EP15 PIC X(4).	POB01460
05 EP16 PIC X(2).	POB01470
05 EP17 PIC X(2).	POB01480
05 EP18 PIC X(2).	POB01490
*	POB01500
*	POB01510
01 VARIABLES-A-GRABAR.	POB01520
* ----SIGUE LA DEFINICION DE LOS CAMPOS DE SALIDA DE CADA VARIABLE	POB01530
05 SP01 PIC X(1).	POB01540
05 SP02 PIC X(1).	POB01550
05 SP03 PIC X(2).	POB01560

05 SNAQUI PIC X(1).	POB01570	
05 SP04A PIC X(1).	POB01580	
05 SP04B PIC X(2).	POB01590	
05 SP04C PIC 9(3).	POB01600	
05 SP04D PIC X(2).	POB01610	
05 SP05 PIC X(3).	POB01620	
05 SP06 PIC X(1).	POB01630	
05 SRAQUI PIC X(1).	POB01640	
05 SP07A PIC X(1).	POB01650	
05 SP07B PIC X(2).	POB01660	
05 SP08 PIC X(1).	POB01670	
05 SP09 PIC X(2).	POB01680	
05 SP10 PIC X(1).	POB01690	
05 SP11 PIC X(1).	POB01700	
05 SP12 PIC X(1).	POB01710	
05 SP13 PIC X(3).	POB01720	
05 SP14 PIC X(1).	POB01730	
05 SP15 PIC X(4).	POB01740	
05 SP16 PIC X(2).	POB01750	
05 SP17 PIC X(2).	POB01760	
05 SP18 PIC X(2).	POB01770	
*	POB01780	
01 OTRAS-VARS.	POB01790	
03 FIN-ARCH	PIC 9 VALUE 0.	POB01800
88 FIN-ARCHIVO	VALUE 1.	POB01810
*	POB01820	
PROCEDURE DIVISION.	POB01830	
* DE LA DATA DIVISION SE PUEDE NOTAR LA SIGUIENTE NOMEMCLATURA	POB01840	
* PARA LA IDENTIFICACION DE CADA VARIABLE DE POBLACION	POB01850	
* -VARIABLE 'VVVV' ES EL NOMBRE ORIGINAL, SE UTILIZA	POB01860	
* COMO NOMBRE DE ARCHIVO Y COMO DDNAME	POB01870	
* -VARIABLE 'RVVVV' ES EL REGISTRO DE GRABACION DE LA	POB01880	
* VARIABLE 'VVVV'	POB01890	
* -VARIABLE 'EVVVV' ES CAMPO DE ENTRADA EN EL REGISTRO	POB01900	
* DEL CENSO	POB01910	
* -VARIABLE 'SVVVV' ES CAMPO DE SALIDA AL CUAL SE LE MUEVE	POB01920	
* LA VARIABLE 'EVVVV' UNA VEZ EDITADA	POB01930	
OPEN INPUT CENSO, OUTPUT	POB01940	
P01	POB01950	
P02	POB01960	
P03	POB01970	
NAQUI	POB01980	
P04A	POB01990	
P04B	POB02000	
P04C	POB02010	
P04D	POB02020	
P05	POB02030	
P06	POB02040	
RAQUI	POB02050	
P07A	POB02060	
P07B	POB02070	
P08	POB02080	
P09	POB02090	

P10	POB02100
P11	POB02110
P12	POB02120
P13	POB02130
P14	POB02140
P15	POB02150
P16	POB02160
P17	POB02170
P18.	POB02180
*	POB02190
READ CENSO INTO REGISTRO-CENSO AT END MOVE 1 TO FIN-ARCH.	POB02200
PERFORM CICLO THRU CICLO-FIN UNTIL FIN-ARCHIVO.	POB02210
DISPLAY " TOTAL REGISTROS LEIDOS ", TOTAL-CENSO.	POB02220
DISPLAY " TOTAL PERSONAS GRABADOS ", TOTAL-PERSONAS.	POB02230
CLOSE CENSO	POB02240
P01	POB02250
P02	POB02260
P03	POB02270
NAQUI	POB02280
P04A	POB02290
P04B	POB02300
P04C	POB02310
P04D	POB02320
P05	POB02330
P06	POB02340
RAQUI	POB02350
P07A	POB02360
P07B	POB02370
P08	POB02380
P09	POB02390
P10	POB02400
P11	POB02410
P12	POB02420
P13	POB02430
P14	POB02440
P15	POB02450
P16	POB02460
P17	POB02470
P18.	POB02480
*	POB02490
STOP RUN.	POB02500
*	POB02510
CICLO.	POB02520
ADD 1 TO TOTAL-CENSO.	POB02530
IF TIPO-REG = 2	POB02540
ADD 1 TO TOTAL-PERSONAS	POB02550
PERFORM PROCESA-PERSONAS.	POB02560
READ CENSO INTO REGISTRO-CENSO AT END MOVE 1 TO FIN-ARCH.	POB02570
CICLO-FIN.	POB02580
EXIT.	POB02590
PROCESA-PERSONAS.	POB02600
EXAMINE REGISTRO-POB REPLACING ALL SPACES BY ZERO.	POB02610
* ESTE EXAMINE ES NECESARIO PUES EL SOFTWARE DE TRANSMISION AL	POB02620

```

* MICROCOMPUTADOR CORTA LOS CARACTERES EN BLANCO QUE APAREZCAN      POB02630
* AL FINAL DE CADA BLOQUE                                           POB02640
  MOVE EP01      TO SP01.                                           POB02650
  MOVE EP02      TO SP02.                                           POB02660
  MOVE EP03      TO SP03.                                           POB02670
  MOVE EP04DX    TO SP04D.                                           POB02680
  IF EP04 = "00008"                                                POB02690
    MOVE "1" TO SNAQUI                                               POB02700
    MOVE PROVINCIA TO SP04A                                           POB02710
    MOVE CANTON   TO SP04B                                           POB02720
    MOVE DISTRITO TO SP04C                                           POB02730
  ELSE                                                                POB02740
    MOVE "0" TO SNAQUI                                               POB02750
    MOVE EP04A TO SP04A                                           POB02760
    MOVE EP04B TO SP04B                                           POB02770
    MOVE EP04C TO SP04C                                           POB02780
    IF EP04A = "0"                                                  POB02790
      MOVE "00"   TO SP04B                                           POB02800
      MOVE EP04Y TO SP04C                                           POB02810
      IF EP04D < 1902                                               POB02820
        MOVE 01 TO SP04D.                                           POB02830
    MOVE EP05      TO SP05.                                           POB02840
    MOVE EP06      TO SP06.                                           POB02850
    MOVE "0"       TO SRAQUI.                                         POB02860
    MOVE EP07A     TO SP07A.                                         POB02870
    MOVE EP07B     TO SP07B.                                         POB02880
    IF EP07 = "008"                                                POB02890
      MOVE PROVINCIA TO SP07A                                         POB02900
      MOVE CANTON   TO SP07B                                         POB02910
      MOVE "1" TO SRAQUI.                                         POB02920
* AQUI SE GREARON LAS VARIABLES NAQUI Y RAQUI: INDICADORES DE      POB02930
* QUE LA PERSONA NACIO EN EL MISMO LUGAR DE EMPADRONAMIENTO Y      POB02940
* QUE LA PERSONA RESIDIA HACE 5 ANNOS EN ESE MISMO LUGAR. ESTO    POB02950
* FUE NECESARIO PORQUE REDATAM, DEPENDIENDO DE LA SELECCION      POB02960
* GEOGRAFICA, NO SIEMPRE PERMITE ACCESAR LAS VARIABLES GEOGRAFIC. POB02970
* ENTONCES SE HACE NECESARIO TENER EN EL REGISTRO LA PROVINCIA,   POB02980
* CANTON Y DISTRITO DE NACIMIENTO Y UN INDICADOR SI ESTE ES IGUAL POB02990
* AL DE ENUMERACION, SIMILARMENTE PARA RESIDENCIA 5 ANNOS ATRAS. POB03000
  MOVE EP08      TO SP08.                                           POB03010
  MOVE EP09      TO SP09.                                           POB03020
  MOVE EP10      TO SP10.                                           POB03030
  MOVE EP11      TO SP11.                                           POB03040
  MOVE EP12      TO SP12.                                           POB03050
  MOVE EP13      TO SP13.                                           POB03060
  MOVE EP14      TO SP14.                                           POB03070
  MOVE EP15      TO SP15.                                           POB03080
  MOVE EP16      TO SP16.                                           POB03090
  MOVE EP17      TO SP17.                                           POB03100
  MOVE EP18      TO SP18.                                           POB03110
  WRITE RP01     FROM SP01.                                           POB03120
  WRITE RP02     FROM SP02.                                           POB03130
  WRITE RP03     FROM SP03.                                           POB03140
  WRITE RNAQUI   FROM SNAQUI.                                         POB03150

```

```

WRITE RP04A FROM SP04A. POB03160
WRITE RP04B FROM SP04B. POB03170
WRITE RP04C FROM SP04C. POB03180
WRITE RP04D FROM SP04D. POB03190
WRITE RP05 FROM SP05. POB03200
WRITE RP06 FROM SP06. POB03210
WRITE RRAQUI FROM SRAQUI. POB03220
WRITE RP07A FROM SP07A. POB03230
WRITE RP07B FROM SP07B. POB03240
WRITE RP08 FROM SP08. POB03250
WRITE RP09 FROM SP09. POB03260
WRITE RP10 FROM SP10. POB03270
WRITE RP11 FROM SP11. POB03280
WRITE RP12 FROM SP12. POB03290
WRITE RP13 FROM SP13. POB03300
WRITE RP14 FROM SP14. POB03310
WRITE RP15 FROM SP15. POB03320
WRITE RP16 FROM SP16. POB03330
WRITE RP17 FROM SP17. POB03340
WRITE RP18 FROM SP18. POB03350
FIN-PROGRAMA. POB03360
EXIT. POB03370
//GO.SYSUDUMP DD SYSOUT=A POB03380
//SYSOUT DD SYSOUT=A POB03390
//CENSO DD UNIT=TAPE,VOL=SER=(R402,R403,R404,R405,R406), POB03400
// DSN=CR84CENS.ORIGIN80 POB03410
//P01 DD UNIT=3350,VOL=SER=OSWORK,DSN=CR84CENS.P01, POB03420
// DCB=(RECFM=FB,LRECL=1,BLKSIZE=19000),DISP=(OLD), POB03430
// SPACE=(TRK,(130,30),RLSE) POB03440
//P02 DD UNIT=3350,VOL=SER=OSWORK,DSN=CR84CENS.P02, POB03450
// DCB=(RECFM=FB,LRECL=1,BLKSIZE=19000),DISP=(OLD), POB03460
// SPACE=(TRK,(130,10),RLSE) POB03470
//P03 DD UNIT=3350,VOL=SER=OSWORK,DSN=CR84CENS.P03, POB03480
// DCB=(RECFM=FB,LRECL=2,BLKSIZE=19000),DISP=(OLD), POB03490
// SPACE=(TRK,(260,20),RLSE) POB03500
//P04A DD UNIT=3350,VOL=SER=OSWORK,DSN=CR84CENS.P04A, POB03510
// DCB=(RECFM=FB,LRECL=1,BLKSIZE=19000),DISP=(OLD), POB03520
// SPACE=(TRK,(130,20),RLSE) POB03530
//P04B DD UNIT=3350,VOL=SER=OSWORK,DSN=CR84CENS.P04B, POB03540
// DCB=(RECFM=FB,LRECL=2,BLKSIZE=19000),DISP=(OLD), POB03550
// SPACE=(TRK,(260,20),RLSE) POB03560
//NAQUI DD UNIT=3350,VOL=SER=OSWORK,DSN=CR84CENS.NAQUI, POB03570
// DCB=(RECFM=FB,LRECL=1,BLKSIZE=19000),DISP=(OLD), POB03580
// SPACE=(TRK,(130,20),RLSE) POB03590
//P04C DD UNIT=3350,VOL=SER=OSWORK,DSN=CR84CENS.P04C, POB03600
// DCB=(RECFM=FB,LRECL=3,BLKSIZE=18000),DISP=(OLD), POB03610
// SPACE=(TRK,(390,10),RLSE) POB03620
//P04D DD UNIT=3350,VOL=SER=OSWORK,DSN=CR84CENS.P04D, POB03630
// DCB=(RECFM=FB,LRECL=2,BLKSIZE=19000),DISP=(OLD), POB03640
// SPACE=(TRK,(260,20),RLSE) POB03650
//P05 DD UNIT=3350,VOL=SER=OSWORK,DSN=CR84CENS.P05, POB03660
// DCB=(RECFM=FB,LRECL=3,BLKSIZE=18000),DISP=(OLD), POB03670
// SPACE=(TRK,(390,10),RLSE) POB03680

```

```

//P06 DD UNIT=3350,VOL=SER=OSWORK,DSN=CR84CENS.P06,          POB03690
// DCB=(RECFM=FB,LRECL=1,BLKSIZE=19000),DISP=(OLD),          POB03700
// SPACE=(TRK,(130,10),RLSE)                                   POB03710
//P07A DD UNIT=3350,VOL=SER=OSWORK,DSN=CR84CENS.P07A,         POB03720
// DCB=(RECFM=FB,LRECL=1,BLKSIZE=19000),DISP=(OLD),          POB03730
// SPACE=(TRK,(130,10),RLSE)                                   POB03740
//P07B DD UNIT=3350,VOL=SER=OSWORK,DSN=CR84CENS.P07B,         POB03750
// DCB=(RECFM=FB,LRECL=2,BLKSIZE=19000),DISP=(OLD),          POB03760
// SPACE=(TRK,(260,10),RLSE)                                   POB03770
//RAQUI DD UNIT=3350,VOL=SER=OSWORK,DSN=CR84CENS.RAQUI,       POB03780
// DCB=(RECFM=FB,LRECL=1,BLKSIZE=19000),DISP=(OLD),          POB03790
// SPACE=(TRK,(130,10),RLSE)                                   POB03800
//P08 DD UNIT=3350,VOL=SER=OSWORK,DSN=CR84CENS.P08,           POB03810
// DCB=(RECFM=FB,LRECL=1,BLKSIZE=19000),DISP=(OLD),          POB03820
// SPACE=(TRK,(130,10),RLSE)                                   POB03830
//P09 DD UNIT=3350,VOL=SER=OSWORK,DSN=CR84CENS.P09,           POB03840
// DCB=(RECFM=FB,LRECL=2,BLKSIZE=19000),DISP=(OLD),          POB03850
// SPACE=(TRK,(260,10),RLSE)                                   POB03860
//P10 DD UNIT=3350,VOL=SER=OSWORK,DSN=CR84CENS.P10,           POB03870
// DCB=(RECFM=FB,LRECL=1,BLKSIZE=19000),DISP=(OLD),          POB03880
// SPACE=(TRK,(130,10),RLSE)                                   POB03890
//P11 DD UNIT=3350,VOL=SER=OSWORK,DSN=CR84CENS.P11,           POB03900
// DCB=(RECFM=FB,LRECL=1,BLKSIZE=19000),DISP=(OLD),          POB03910
// SPACE=(TRK,(130,10),RLSE)                                   POB03920
//P12 DD UNIT=3350,VOL=SER=OSWORK,DSN=CR84CENS.P12,           POB03930
// DCB=(RECFM=FB,LRECL=1,BLKSIZE=19000),DISP=(OLD),          POB03940
// SPACE=(TRK,(130,10),RLSE)                                   POB03950
//P13 DD UNIT=3350,VOL=SER=OSWORK,DSN=CR84CENS.P13,           POB03960
// DCB=(RECFM=FB,LRECL=3,BLKSIZE=18000),DISP=(OLD),          POB03970
// SPACE=(TRK,(390,10),RLSE)                                   POB03980
//P14 DD UNIT=3350,VOL=SER=OSWORK,DSN=CR84CENS.P14,           POB03990
// DCB=(RECFM=FB,LRECL=1,BLKSIZE=19000),DISP=(OLD),          POB04000
// SPACE=(TRK,(130,10),RLSE)                                   POB04010
//P15 DD UNIT=3350,VOL=SER=OSWORK,DSN=CR84CENS.P15,           POB04020
// DCB=(RECFM=FB,LRECL=4,BLKSIZE=19000),DISP=(OLD),          POB04030
// SPACE=(TRK,(520,10),RLSE)                                   POB04040
//P16 DD UNIT=3350,VOL=SER=OSWORK,DSN=CR84CENS.P16,           POB04050
// DCB=(RECFM=FB,LRECL=2,BLKSIZE=19000),DISP=(OLD),          POB04060
// SPACE=(TRK,(260,10),RLSE)                                   POB04070
//P17 DD UNIT=3350,VOL=SER=OSWORK,DSN=CR84CENS.P17,           POB04080
// DCB=(RECFM=FB,LRECL=2,BLKSIZE=19000),DISP=(OLD),          POB04090
// SPACE=(TRK,(260,10),RLSE)                                   POB04100
//P18 DD UNIT=3350,VOL=SER=OSWORK,DSN=CR84CENS.P18,           POB04110
// DCB=(RECFM=FB,LRECL=2,BLKSIZE=19000),DISP=(OLD),          POB04120
// SPACE=(TRK,(260,10),RLSE)                                   POB04130

```

1

2

3

4



