

IMPLEMENTACIÓN DEL PROCEDIMIENTO DE IMPUTACIÓN HOTDECK (Variables Cualitativas)

1. INSTRUCCIONES GENERALES

Para la implementación del procedimiento de imputación HOTDECK, se debe precisar lo siguiente:

- La imputación es efectuada sólo a miembro del hogar, (de miembro a miembro).
- La imputación se efectúa sólo a miembros del hogar que tiene omitidos todos los capítulos que les corresponde (Capítulos: Educación, Salud y Empleo)
 - a. Generar la población a ser imputada (OMISOS).
 - Miembros del hogar que tengan omitidos todos los capítulos que les corresponden.
 - b. Generar la población que proporcionará los datos (DONANTES).
 - Miembros del hogar que tengan los capítulos que les corresponde completos.

La imputación sobre las variables cualitativas faltantes se efectúa mediante el reemplazo de las características de información cualitativa de las personas que respondieron los módulos de educación (300), módulo de salud (400) y empleo (500) con características demográficas, educativas, espaciales, socioeconómicas y temporales similares, a estas personas se les denomina "DONANTES".

2. PROCESO DE IMPUTACIÓN.

Capítulo 300 "Educación"

Se Remplazan todas las variables de la persona "Omisa" con las variables de la persona "Donante". Para las variables cuantitativas, se debe cambiar el valor de estas por un valor missing (999999).

Variables que deben cambiar a valor missing (999999)

- Si $P311(1:7)=1 \wedge P311A(1) = 1 \Rightarrow P311B(1) = \text{Missing}$
- Si $P311(1:7)=1 \wedge P311A(2:7) = 1 \Rightarrow P311D(2:7) = \text{Missing}$
- Si $P312(1:2)=1 \wedge P312A(1) = 1 \Rightarrow P311B(1) = \text{Missing}$
- Si $P312(1:1)=1 \wedge P312A(2:6) = 1 \Rightarrow P311C(2:6) = \text{Missing}$
- Si $P314=1 \wedge P314(1) = 1 \Rightarrow P314B = \text{Missing}$
- Si $P314=1 \wedge P314(2:6) = 1 \Rightarrow P314C = \text{Missing}$

Capítulo 400 “Salud”

Se Remplazan todas las variables de la persona “Omisa” con las variables de la persona “Donante”. Para las variables cuantitativas, se debe cambiar el valor de estas por un valor missing (999999).

Variables que deben cambiar a valor missing (999999)

- Si $P414(1:16)=1 \wedge P415(1,9) = 1 \Rightarrow P416(1:16) = \text{Missing}$
- Si $P414(1:16)=1 \wedge P415(2:8) = 1 \Rightarrow P418(1:16) = \text{Missing}$

Capítulo 500 “Empleo”

Se Remplazan todas las variables de la persona “Omisa” con las variables de la persona “Donante”. Para las variables cuantitativas, se debe cambiar el valor de estas por un valor missing (999999).

Variables que deben cambiar a valor missing (999999)

Ingresos, para trabajador dependiente – actividad principal.

- Si $p513T \geq 0 \vee p513T = \text{Missing} \Rightarrow P513T = \text{Missing}$
- Si $P524A > 0 \vee P524A = \text{Missing} \Rightarrow P524A = \text{Missing}$
- Si $P524B > 0 \vee P524B = \text{Missing} \Rightarrow P524B = \text{Missing}$
- $P524C = 0$
- $P524D = 0$
- Si $P524E > 0 \vee P524E = \text{Missing} \Rightarrow P524E = \text{Missing}$
- Si $P5294B > 0 \vee P5294B = \text{Missing} \Rightarrow P5294 = \text{Missing}$

Ingresos, para trabajador dependiente – actividad secundaria.

- Si $p518 \geq 0 \vee p518 = \text{Missing} \Rightarrow P518 = \text{Missing}$
- Si $P538A > 0 \vee P538A = \text{Missing} \Rightarrow P538A = \text{Missing}$
- Si $P538B > 0 \vee P538B = \text{Missing} \Rightarrow P538B = \text{Missing}$
- $P538C = 0$
- $P538D = 0$
- Si $P538E > 0 \vee P538E = \text{Missing} \Rightarrow P538E = \text{Missing}$
- Si $P5404B > 0 \vee P5404B = \text{Missing} \Rightarrow P5404 = \text{Missing}$

Gastos en alimentos consumidos fuera del hogar

- Si $P559(i) = 1 \wedge (P559D(i) > 0 \vee (\text{Missing})) \Rightarrow P559D = \text{Missing}$

Otros Gastos

- Si $P560(i) = 1 \wedge (P560D(i) > 0 \vee (\text{Missing})) \Rightarrow P560D = \text{Missing}$

3. JERARQUÍA DE VARIABLES PARA SELECCIÓN DE DONANTES.

Para efectuar el proceso de imputación (de Donantes a Omisos), se divide la población de omisos en 3 sub poblaciones, tanto para donantes como para Omisos, tomando como variables discriminante la variable P203 “Relación de parentesco”.

SUB POBLACIONES

- **Sub Población 1:** Para imputar Jefes del Hogar Omisos (P203= 1), se debe obtener los Donantes de una población de jefes de Hogar P203=1.
- **Sub Población 2:** Para Imputar cónyuges Omisos (esposos o esposas, P203=2), se debe obtener los Donantes de una población de conyugues (esposos o esposas, P203=2)
- **Sub Población 3:** Para imputar al resto de miembros del hogar Omisos (hijos, yernos nueras, nietos, padres o suegros, otros parientes y otros no parientes, P203 = 3, 4, 5, 6, 7, 10), se debe obtener los Donantes de una población de (hijos, yernos nueras, nietos, padres o suegros, otros parientes y otros no parientes, P203 = 3, 4, 5, 6, 7, 10).

3.1. SELECCIÓN DE OMISOS Y DONANTES PARA JEFES DEL HOGAR.

- f Sub población de jefes de hogar omisos deben cumplir las siguientes características simultáneamente:

P203 = 1 “Jefe del Hogar”
Omiso en el Cap300 “Educación”
Omiso en el Cap400 “Salud”
Omiso en el Cap500 “Empleo”

- f Sub población de jefes de hogar Donantes, deben cumplir las siguientes características simultáneamente:

P203 = 1 “Jefe del Hogar”
No Omiso en el Cap300 “Educación”
No Omiso en el Cap400 “Salud”
No Omiso en el Cap500 “Empleo”

Para realizar la imputación, cada Omiso y Donante, debe encontrarse en la siguiente jerarquía de variables.

1. Dominio (departamento): Dominio
2. Estrato socioeconómico: Estratos
3. Edad: P208A
4. Sexo: P207
5. Número de miembros por hogar: MIEPERHO
6. Periodo (trimestre): TRIMEST

DESCRIPCIÓN DE LAS VARIABLES JERÁRQUICAS

1. Dominio.
Se considera como dominio a cada uno de los departamentos del país.
2. Estrato socioeconómico.
La variable estrato socioeconómico, toma los siguientes valores:
1 : Estrato A
2 : Estrato B

- 3 : Estrato C
- 4 : Estrato D
- 5 : Resto urbano
- 6 : Rural

3. Edad.

La variable edad debe considerar los siguientes rangos:

- 1 : 18 a 24 años.
- 2 : de 25 a 50 años.
- 3 : de 51 a 64 años.
- 4 : de 65 años a más.

4. Sexo.

La variable sexo toma los siguientes valores:

- 1 : Hombre
- 2 : Mujer

5. Número de miembros por hogar.

La variable miembros por hogar debe tomar los siguientes rangos:

- 1: De 1 a 2 miembros.
- 2: De 3 a 4 miembros.
- 3: De 5 a 6 miembros.
- 4: De 7 a 8 miembros.
- 5: De 9 a más miembros.

6. Periodo (Trimestre).

La variable periodo, debe considerar los siguientes valores:

- 1 : Enero, Febrero y Marzo
- 2 : Abril, Mayo y Junio
- 3 : Julio, Agosto y Septiembre
- 4 : Octubre, Noviembre y Diciembre

3.2. SELECCIÓN DE OMISOS Y DONANTES PARA CÓNYUGES (ESPOSOS, ESPOSAS).

Sub población de Cónyuges omisos, deben cumplir las siguientes características simultáneamente:

- P203 = 2 "Cónyuge"
- Omiso en el Cap300 "Educación"
- Omiso en el Cap400 "Salud"
- Omiso en el Cap500 "Empleo"(*)
- (*) Considerando la Edad del Informante

Sub población de Cónyuges Donantes, deben cumplir las siguientes características simultáneamente:

- P203 = 2 "Cónyuges"
- No Omiso en el Cap300 "Educación"
- No Omiso en el Cap400 "Salud"
- No Omiso en el Cap500 "Empleo"(*)
- Nivel del Jefe del Hogar != Missing
- (*) Considerando la Edad del Informante

Para realizar la imputación cada Omiso y Donante, debe encontrarse en la siguiente jerarquía de variables.

1. Dominio (departamento) : Dominio
2. Estrato socioeconómico: Estratos
3. Edad: P208A
4. Sexo: P207
5. Número de miembros por hogar: MIEPERHO
6. Nivel de educación del jefe del hogar: P301A
7. Periodo (trimestre) : TRIMEST

DESCRIPCIÓN DE LAS VARIABLES JERÁRQUICAS

1. Dominio.
Se considera como dominio a cada uno de los departamentos del país.
2. Estrato socioeconómico.
La variable estrato socioeconómico, toma los siguientes valores:
1 : Estrato A
2 : Estrato B
3 : Estrato C
4 : Estrato D
5 : Resto urbano
6 : Rural
3. Edad.
La variable edad debe considerar los siguientes rangos:
1 : 12 a 14 años.
2 : 15 a 18 años.
3 : 19 a 24 años.
4 : de 25 a 50 años.
5 : de 51 a 64 años.
6 : de 65 años a más.
4. Sexo.
La variable sexo toma los siguientes valores:
1 : Hombre
2 : Mujer
5. Número de miembros por hogar.
La variable miembros por hogar debe tomar los siguientes rangos:
1: De 1 a 2 miembros.
2: De 3 a 4 miembros.
3: De 5 a 6 miembros.
4: De 7 a 8 miembros.
5: De 9 a más miembros.
6. Nivel educativo del jefe del hogar.
La variable jefe del hogar debe tomar los siguientes valores:
1 : Primaria (incluye inicial y sin nivel).
2 : Secundaria.
3 : Superior.

7. Periodo (Trimestre).

La variable periodo, debe considerar los siguientes valores:

- 1 : Enero, Febrero y Marzo
- 2 : Abril, Mayo y Junio
- 3 : Julio, Agosto y Septiembre
- 4 : Octubre, Noviembre y Diciembre

3.3. SELECCIÓN DE DON ANTES PARA RESTO DE MIEMBROS DEL HOGAR (HIJO, YERNO NUERA, NIETOS, PADRES O SUEGROS, OTROS PARIENTES Y OTROS NO PARIENTES).

Sub población de otros miembros del hogar (hijo, yerno nuera, nietos, padres o suegros, otros parientes y otros no parientes) omisos, deben cumplir las siguientes características simultáneamente:

P203 = 3 ∨ 4 ∨ 5 ∨ 6 ∨ 7 ∨ 10 “hijo, yerno nuera, nietos, padres o suegros, otros parientes y otros no parientes”

Omiso en el Cap300 “Educación”

Omiso en el Cap400 “Salud”

Omiso en el Cap500 “Empleo”

f Sub población de otros miembros del hogar (hijo, yerno nuera, nietos, padres o suegros, otros parientes y otros no parientes) Donantes, deben cumplir las siguientes características simultáneamente:

P203 = 3 ∨ 4 ∨ 5 ∨ 6 ∨ 7 ∨ 10 “hijo, yerno nuera, nietos, padres o suegros, otros parientes y otros no parientes”

No Omiso en el Cap300 “Educación”

No Omiso en el Cap400 “Salud”

No Omiso en el Cap500 “Empleo”

Para realizar la imputación cada Omiso y Donante, debe encontrarse en la siguiente jerarquía de variables.

1. Dominio (departamento): Dominio
2. Estrato socioeconómico: Estratos
3. Edad: P208A
4. Sexo: P207
5. Número de miembros por hogar: MIEPERHO
6. Nivel de educación del jefe del hogar: P301A
7. Periodo (trimestre): TRIMEST

DESCRIPCIÓN DE LAS VARIABLES JERÁRQUICAS

1. Dominio.

Se considera como dominio a cada uno de los departamentos del país.

2. Estrato socioeconómico.

Las variables estrato socioeconómico, toma los siguientes valores:

- 1 : Estrato A
- 2 : Estrato B
- 3 : Estrato C
- 4 : Estrato D
- 5 : Resto urbano
- 6 : Rural

3. Edad.

La variable edad debe considerar los siguientes rangos:

- 1 : Menor de 1 año.
- 2 : de 1 a 4 años
- 3 : 5 años.
- 4 : 6 años.
- 5 : 7 años.
- 6 : 8 años.
- 7 : 9 años.
- 8 : 10 años.
- 9 : 11 años.
- 10 : 12 años.
- 11 : 13 años.
- 12 : 14 años.
- 13 : 15 años.
- 14 : 16 años.
- 15 : 17 años.
- 16 : 18 años.
- 17 : 19 años.
- 18 : 20 años.
- 19 : 21 años.
- 20 : 22 años.
- 21 : 23 años.
- 22 : 24 años.
- 23 : de 25 a 50 años.
- 24 : de 51 a 64 años.
- 25 : de 65 años a más.

4. Sexo.

La variable sexo toma los siguientes valores:

- 1 : Hombre
- 2 : Mujer

5. Número de miembros por hogar.

La variable miembros por hogar debe tomar los siguientes rangos:

- 1: De 1 a 2 miembros.

- 2: De 3 a 4 miembros.
- 3: De 5 a 6 miembros.
- 4: De 7 a 8 miembros.
- 5: De 9 a más miembros.

6. Nivel educativo del jefe del hogar.

La variable jefe del hogar debe tomar los siguientes valores:

- 1 : Primaria (incluye inicial y sin nivel).
- 2 : Secundaria.
- 3 : Superior.

7. Periodo (Trimestre).

La variable periodo, debe considerar los siguientes valores:

- 1 : Enero, Febrero y Marzo
- 2 : Abril, Mayo y Junio
- 3 : Julio, Agosto y Septiembre
- 4 : Octubre, Noviembre y Diciembre

4. IMPUTACIÓN

Imputar miembro a miembro (de Donante a Omiso) para todos aquellos casos en los que la siguiente cadena de variables (Omiso) sea similar a la cadena de variables (Donante)

Para jefes de Hogar

Cadena de Variables OMISOS		Cadena de Variables DONANTES
Dominio + Estrato + Edad + Sexo + Nro. de Miembros del Hogar + Periodo	=	Dominio + Estrato + Edad + Sexo + Nro de Miembros del Hogar + Periodo

Para Cónyuges y Resto de miembros del Hogar

Cadena de Variables OMISOS		Cadena de Variables DONANTES
Dominio + Estrato + Edad + Sexo + Nro de Miembros del Hogar + Nivel del Jefe del Hogar + Periodo	=	Dominio + Estrato + Edad + Sexo + Nro de Miembros del Hogar + Nivel del Jefe del Hogar + Periodo

5. OBSERVACIONES

La selección de donantes en una lista de donantes, se debe seguir el ordenamiento con las siguientes especificaciones:

Urbano: Ubigeo+zona+manzana+Conglomerado+vivienda+hogar+Nro de persona

Rural: Ubigeo+Aer+Conglomerado+vivienda+Hogar+Nro de persona

De faltar donantes en una primera búsqueda, abrir en primer orden dentro de la jerarquía, la variable PERIODO. En segundo orden, la variable ESTRATO.

ANEXO

Introducción

El procesamiento tradicional ENAH0 contempla:

- a) La deflactación de valores monetarios.
- b) La imputación de variables (horas trabajadas, ingresos, gastos).
- c) El cálculo de la sumaria.

Adicionalmente a estos procesos, se ha incorporado un proceso de imputación HotDeck (después de la deflactación y antes de la imputación) para las omisiones de personas en el cuestionario ENAH0.01A, como es de esperarse los módulos afectados son el 300, 400 y 500.

Este proceso HotDeck parte de una identificación de receptores y donantes que cumplan con los criterios establecidos para cada grupo de parentesco, se definen (3) tres: Jefes de Hogar, Cónyuges y el Resto (Hijo, Yerno, Nuera, Nietos, Padres, Suegros, Otros parientes, otros no parientes). *Para una descripción detallada de los criterios establecidos, por favor revise cualquiera de los documentos: “Procedimiento de Imputación Hot-Deck 2004”, “Procedimiento de Imputación Hot-Deck 2005” o “Procedimiento de Imputación Hot-Deck 2006”*

Encontrado el donante correspondiente a un receptor, los datos cualitativos de este donante reemplazan los datos cualitativos del receptor. Los datos cuantitativos del donante son cambiados a “missing” en el receptor. Completado el proceso HotDeck, el procesamiento continúa con el proceso tradicional de imputación de datos.

Para efectos de asignación de un donante, los datos son ordenados de acuerdo al siguiente grupo de variables:

- Estrato Socio-Económico
- Edad (Grupo de edad).
- Sexo.
- N° de miembros del hogar.
- Nivel educativo del Jefe del Hogar.
- Período (Trimestre al que pertenece la encuesta).
- UBIGEO (Código de Ubicación Geográfica).
- Zona (Identificación Censal).
- Manzana (Identificación Censal).
- Suf-Manzana (Identificación Censal).
- AER (Identificación Censal).
- Conglomerado.
- Vivienda.
- Hogar.
- Persona.

I. Inventario y Código Fuentes de Programas

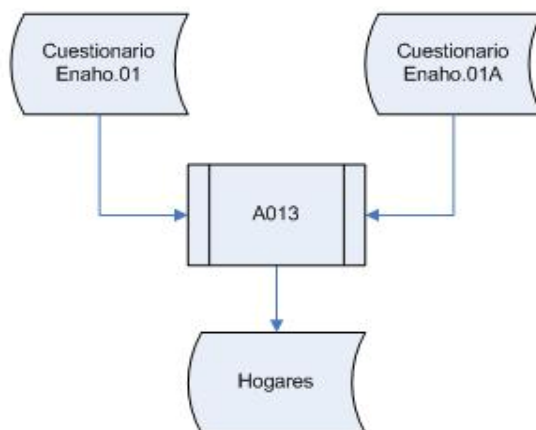
A013	Generación de variables del hogar.
A001	Identificación y generación de Jefes del Hogar Receptores.
A006	Identificación y generación de Jefes del Hogar Donantes.
A002	Identificación y generación de Cónyuges Receptores.
A007	Identificación y generación de Cónyuges Donantes.
A003	Identificación y generación de Resto Receptores.
A008	Identificación y generación de Resto Donantes.
AsigDona	Establece Donante-Receptor en 3 iteraciones.
A009	Imputación Hot-Deck.

Los programas A013, A001, A006, A002, A007, A003, A008, A009 están implementados en ISSA-X, en cambio AsigDona lo está en Foxpro 2.6.

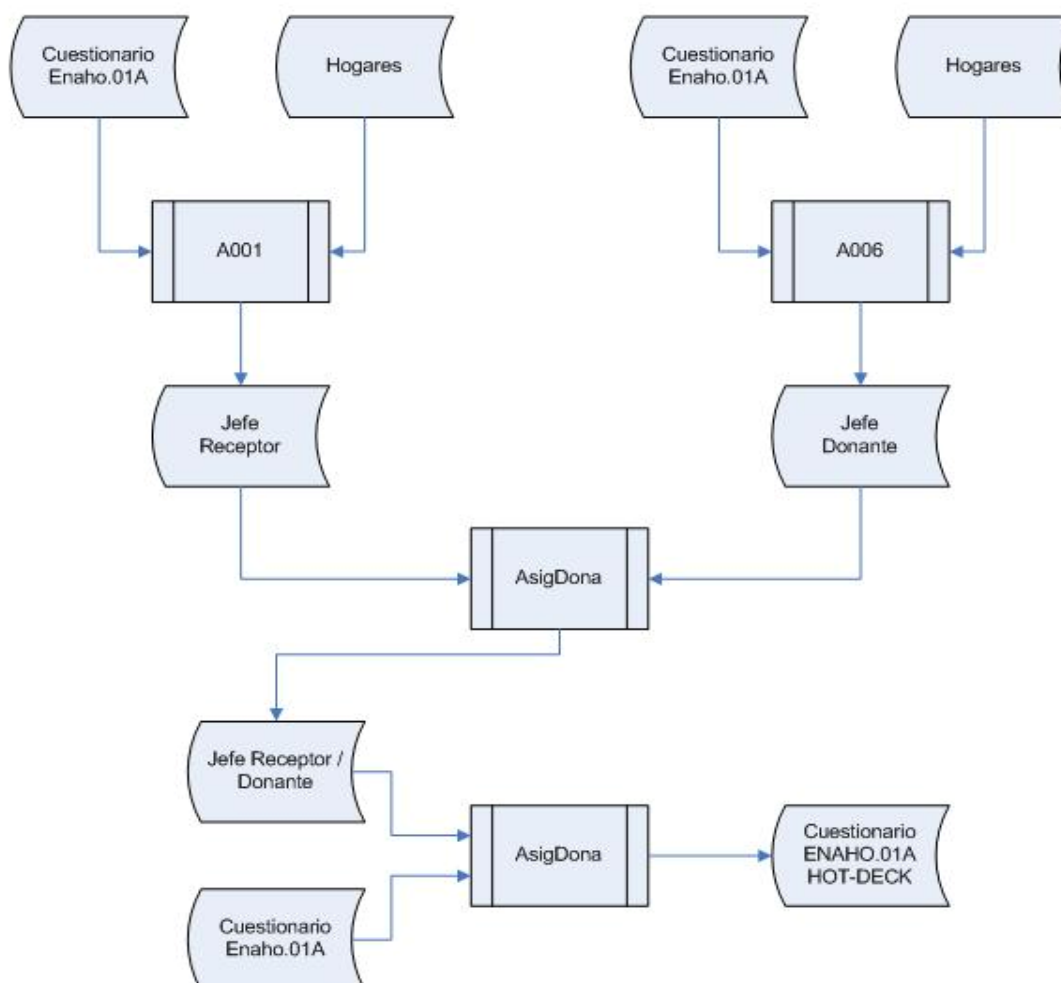
II. Diagrama General del Proceso de Imputación Hot-Deck

IMPUTACION HOT-DECK

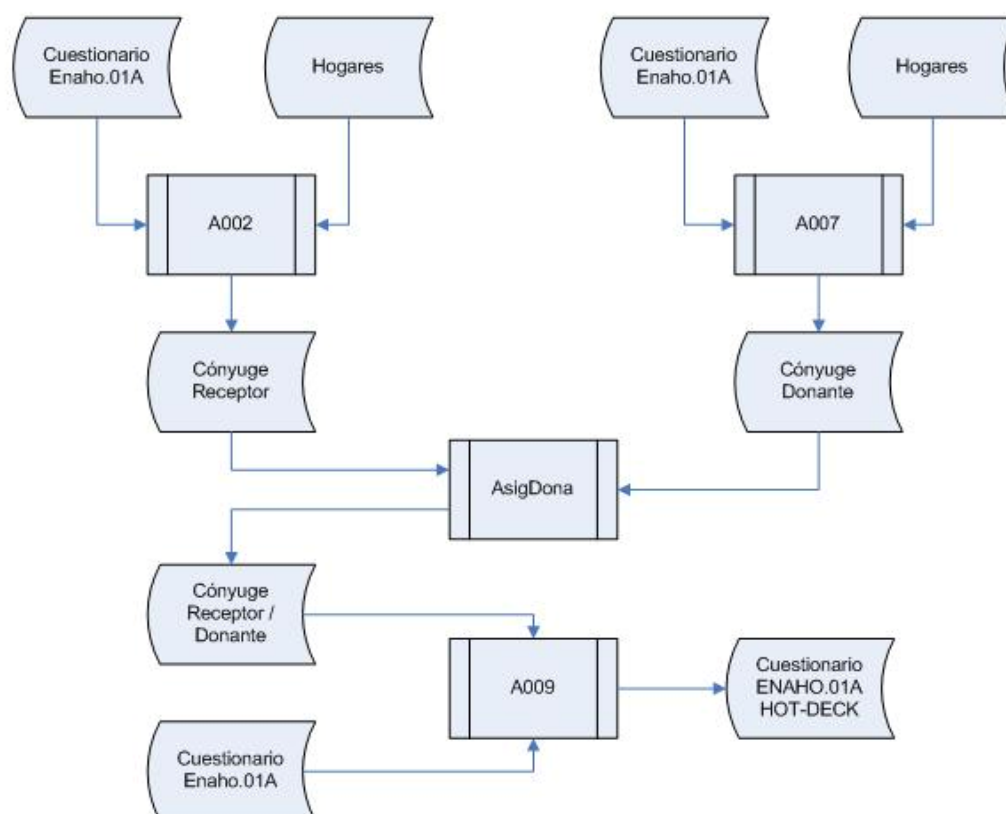
Paso (1) : Generando variables preliminares de Hogar.



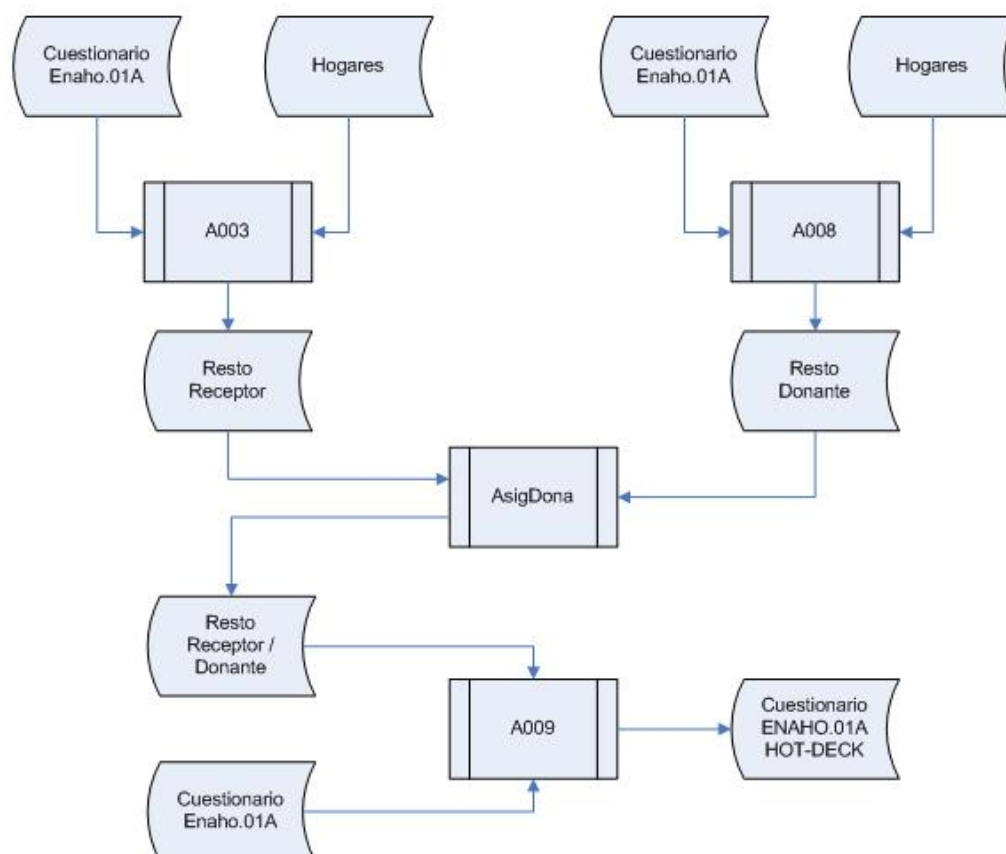
Paso (2) : Imputación Hot-Deck para el Jefe del Hogar.



Paso (3) : Imputación Hot-Deck para el Cónyuge.



Paso (4) : Imputación Hot-Deck para el Resto.



III. Programas Fuentes

Programa A013

Genera las variables por hogar necesarias para los procesos de imputación posteriores.
Estas variables son las siguientes:

- Ubigeo.
- Zona.
- Manzana.
- SufManza.
- Aer.
- Nivel.
- Miembros.
- Estrato.

Código

PROC EA03401A

PreProc

Level 1

If par300 <> 1 Then

 Skip Case;

Endif;

PostProc

Level 1

Persona=NotAppl;

r=LoadCase(EA034001,pConglo,pCodViv,pHog1,pHog2,Persona);

If r=1 Then

 mieperho=0;

 i=1;

 n200=soccurs(cap200);

 While i <= n200 Do;

 If p204(i) = 1 and p203(i) <> 8 and p203(i) <> 9 then

 mieperho = mieperho + 1;

 Endif;

 i=i+1;

 Enddo;

Estrato=0;

r=LoadCase(EstraSoc,pConglo);

If r=1 Then

 Estrato=EstSocio;

Endif;

```
Miembros = mieperho;  
Nivel    = p301a;  
Zona     = Zona01[1:3];  
Manzana  = "      ";  
SufManza = "      ";
```

```
If Poschar( Manz01[4:1], 'ABCDEFGHIJKLMNOPQRSTUVWXYZ') <> 0 then  
    Manzana = Manz01[1:3];  
    SufManza = Manz01[4:1];  
Endif;
```

```
If Poschar( Manz01[4:1], '0123456789') <> 0 then  
    Manzana = Manz01[2:3];  
    SufManza = " ";  
Endif;
```

```
Hogar = Concat(  
Edit('ZZZ9',pConglo),Edit('ZZ9',pCodViv),Edit('9',pHog1),Edit('9',pHog2),'1' );  
Export format record(  
Hogar,pConglo,pCodViv,pHog1,pHog2,Ubigeo01,Zona,Manzana,SufManza,Aer1,Nive  
l,Miembros,Estrato );  
  
Endif;
```

Programa A001

Identifica y genera la relación de Jefes de Hogar Receptores.

Código

PROC A001

Function F_JefeOmiso()

```
Miembro = ( mie300=1 & par300 <> 8 & par300 <> 9 ) &  
           ( mie400=1 & par400 <> 8 & par400 <> 9 ) &  
           ( mie500=1 & par500 <> 8 & par500 <> 9 );
```

```
Omiso=( p300i=0 & par300=1 ) & ( p400i=0 & par400=1 ) & ( p500i=0 & par500=1 );
```

```
F_JefeOmiso = ( Miembro & Omiso );
```

End;

Function F_Miembros(Cod)

```
Box Cod    => Ok;  
1-2  => 1;  
3-4  => 2;  
5-6  => 3;  
7-8  => 4;  
>=9 => 5;
```


EndBox;

F_Miembros = Ok;

End;

Function F_Edad(Cod)

Box Cod => Ok;

18-24 => 1;

25-50 => 2;

51-64 => 3;

65-98 => 4;

EndBox;

F_Edad = Ok;

End;

Function F_Periodo(Cod)

Box Cod => Ok;

1-3 => 1;

4-6 => 2;

7-9 => 3;

10-12 => 4;

EndBox;

F_Periodo=Ok;

End;

PROC EA03401A

Level 1

If F_JefeOmiso() Then

r=LoadCase(Hogares,pConglo,pCodViv,pHog1,pHog2);

If r=1 Then

Miembros = F_Miembros(HMiembro);

Edad = F_Edad(Eda300);

Periodo = F_Periodo(ToNumber(Mes_01A));

Nivel = NotAppl;

Export Format Record(pConglo, pCodViv, pHog1, pHog2, pPer, HEstrato, Edad,
Sex300, Miembros, Nivel, Periodo, HUbigeo, HZona, HManza, HSufManz, HAer);

Endif;

Endif;

Programa A006

Identifica y genera la relación de Donantes para el Jefe de Hogar.

Código

PROC A006

Function F_JefeDona()

Dona300 = (p300i > 0 & par300=1 & mie300=1 & par300 <> 8 & par300 <> 9 & eda300 >= 18);

Dona400 = (p400i > 0 & par400=1 & mie400=1 & par400 <> 8 & par400 <> 9 & edaa400 >= 18);

Dona500 = (p500i > 0 & par500=1 & mie500=1 & par500 <> 8 & par500 <> 9 & eda500 >= 18);

F_JefeDona = (Dona300 & Dona400 & Dona500);

End;

Function F_Edad(Cod)

Ok=0;

Box Cod => Ok;

18-24 => 1;

25-50 => 2;

51-64 => 3;

65-98 => 4;

Endbox;

F_Edad = Ok;

End;

Function F_MiePerHo(Cod)

Ok=0;

Box Cod => Ok;

1-2 => 1;

3-4 => 2;

5-6 => 3;

7-8 => 4;

>=9 => 5;

EndBox;

F_MiePerHo=Ok;

End;

Function F_Periodo(Cod)

```
Ok=0;
Box Cod => Ok;
  1-3  => 1;
  4-6  => 2;
  7-9  => 3;
  10-12 => 4;
EndBox;
```

```
F_Periodo=Ok;
```

End;

PROC EA03401A

PreProc

Level 0

Casos=0;

Level 1

```
r=LoadCase( Hogares,pConglo,pCodViv,pHog1,pHog2 );
```

```
If r=0 Then
```

```
  Skip Case;
```

```
Else
```

```
  If (p300i=0 & p400i=0 & p500i=0) Or (p300i=NotAppl &      p400i=NotAppl &
    p500i=NotAppl) Then
```

```
    Skip Case;
```

```
  Endif;
```

```
Endif;
```

PostProc

Level 1:

```
If F_JefeDona()      Then
```

```
  Mes    = ToNumber( Mes_01A );
```

```
  Sexo   = Sex300;
```

```
  Edad   = F_Edad( Eda300 );
```

```
  Miembros = F_MiePerHo( HMiembro );
```

```
  Periodo = F_Periodo( Mes );
```

```
  Nivel   = NotAppl;
```

```
  Export Format Record( pConglo, pCodViv, pHog1, pHog2, pPer, HEstrato, Edad,
Sexo, Miembros,Nivel,Periodo,HUbigeo,HZona,HManza,HSufmanz,HAer );
```

```
Endif;
```

Programa A002

Identifica y genera la relación de Cónyuges Receptores.

Código

PROC A002

Function F_ConyugeOmiso()

```
Omiso = ( ( eda300 >= 12 & eda300 < 14 & p300i = 0 ) & ( edaa400 >= 12 &
edaa400 < 14 & p400i = 0 ) ) Or
( ( eda300 >= 14 & eda300 <= 98 & p300i = 0 ) & ( edaa400 >= 14 & edaa400
<= 98 & p400i = 0 ) & ( eda500 >= 14 & eda500 <= 98 & p500i = 0 ) );
```

```
Conyuge = ( par300 = 2 & par400 = 2 ) Or
( par300 = 2 & par400 = 2 & par500 = 2 );
```

```
Miembro = ( mie300=1 & par300 <> 8 & par300 <> 9 & mie400=1 & par400 <> 8 &
par400 <> 9 ) Or
( mie300=1 & par300 <> 8 & par300 <> 9 & mie400=1 & par400 <> 8 &
par400 <> 9 & mie500=1 & par500 <> 8 & par500 <> 9 );
```

```
F_ConyugeOmiso = Omiso & Conyuge & Miembro;
```

End;

Function F_Miembros(Cod)

```
Box Cod => Ok;
```

```
1-2 => 1;
```

```
3-4 => 2;
```

```
5-6 => 3;
```

```
7-8 => 4;
```

```
>=9 => 5;
```

```
EndBox;
```

```
F_Miembros = Ok;
```

End;

Function F_Edad(Cod)

```
Box Cod => Ok;
```

```
10-14 => 1;
```

```
15-18 => 2;
```

```
19-24 => 3;
```

```
25-50 => 4;
```

```
51-64 => 5;
```

```
>= 65 => 6;
```

```
EndBox;
```

F_Edad = Ok;

End;

Function F_Periodo(Cod)

Box Cod => Ok;

1-3 => 1;

4-6 => 2;

7-9 => 3;

10-12 => 4;

EndBox;

F_Periodo=Ok;

End;

Function F_NivelA(Cod)

Box Cod => Ok;

1-2 => 1;

3-4 => 2;

5-6 => 3;

7-11 => 4;

EndBox;

F_NivelA=Ok;

End;

Function F_NivelB(Cod)

Box Cod => Ok;

1-4 => 1;

5-6 => 2;

7-11 => 3;

EndBox;

F_NivelB=Ok;

End;

Function F_NivelC(Cod)

Box Cod => Ok;

1-6 => 1;

7-11 => 2;

EndBox;

F_NivelC=Ok;

```
End;
PROC EA03401A
PreProc
Level 0
Parametr = ToNumber( SysParm() );

Level 1

If par300 <> 2 Or par400 <> 2 Or Par500 <> 2 Then
  Skip Case;
Endif;

r=LoadCase( Hogares,pConglo,pCodViv,pHog1,pHog2 );

If r=1 Then
  If HNivel=Missing Or HNivel=0 Or HNivel=NotAppl Then
    Skip Case;
  Endif;
Else
  Skip Case;
Endif;

PostProc
Level 1

If F_ConyugeOmiso() Then

  Ed = 0;

  If Ed=0 & Eda300 > 0 Then
    Ed=Eda300;
  Endif;

  If Ed=0 & Edaa400 > 0 Then
    Ed=Edaa400;
  Endif;

  If Ed=0 & Eda500 > 0 Then
    Ed=Eda500;
  Endif;

  Sexo=0;

  If Sexo=0 & Sex300 > 0 Then
    Sexo=Sex300;
  Endif;

  If Sexo=0 & Sex400 > 0 Then
    Sexo=Sex400;
```

Endif;

If Sexo=0 & Sex500 > 0 Then

 Sexo=Sex500;

Endif;

Miembros = F_Miembros(HMiembro);

Edad = F_Edad(Ed);

Periodo = F_Periodo(ToNumber(Mes_01A));

If Parametr = 1 Then

 Nivel = F_NivelA(HNivel);

Endif;

If Parametr = 2 Then

 Nivel = F_NivelB(HNivel);

Endif;

If Parametr = 3 Then

 Nivel = F_NivelC(HNivel);

Endif;

{ --- Exporta Conyuge Omiso --- }

Export Format Record(pConglo, pCodViv, pHog1, pHog2, pPer, HEstrato, Edad,
Sexo, Miembros, Nivel, Periodo, HUbigeo, HZona, HManza, HSufManz, HAer);

Endif;

Programa A007

Identifica y genera la relación de Cónyuges Donantes.

Código

PROC A007

Function F_ConyugeDona()

Completo = ((eda300 >= 12 & eda300 < 14 & p300i > 0) & (edaa400 >= 12 &
 edaa400 < 14 & p400i > 0)) Or
 ((eda300 >= 14 & eda300 <= 98 & p300i > 0) & (edaa400 >= 14 &
 edaa400 <= 98 & p400i > 0) & (eda500 >= 14 & eda500 <= 98 & p500i > 0
));

Conyuge = (par300 = 2 & par400 = 2) Or (par300 = 2 & par400 = 2 & par500 = 2);

Miembro = (mie300=1 & par300 <> 8 & par300 <> 9 & mie400=1 & par400 <> 8 &
par400 <> 9) Or (mie300=1 & par300 <> 8 & par300 <> 9 & mie400=1 & par400 <>
8 & par400 <> 9 & mie500=1 & par500 <> 8 & par500 <> 9);

```
Mayor12 = ( ( eda300 >= 12 & eda300 < 14 ) & ( edaa400 >= 12 & edaa400 < 14 ) )  
Or ( ( eda300 >= 14 & eda300 <= 98 ) & ( edaa400 >= 14 & edaa400 <= 98 ) & ( eda500 >= 14 & eda500 <= 98 ) );
```

```
F_ConyugeDona = Completo & Conyuge & Miembro & Mayor12;
```

```
End;
```

```
Function F_Edad( Cod )
```

```
Ok=0;  
Box Cod => Ok;  
10-14 => 1;  
15-18 => 2;  
19-24 => 3;  
25-50 => 4;  
51-64 => 5;  
65-98 => 6;  
Endbox;
```

```
F_Edad=Ok;
```

```
End;
```

```
Function F_Miembros( Cod )
```

```
Ok=0;  
Box Cod => Ok;  
1-2 => 1;  
3-4 => 2;  
5-6 => 3;  
7-8 => 4;  
>=9 => 5;  
EndBox;
```

```
F_Miembros=Ok;
```

```
End;
```

```
Function F_Periodo( Cod )
```

```
Ok=0;  
Box Cod => Ok;  
1-3 => 1;  
4-6 => 2;  
7-9 => 3;  
10-12 => 4;  
EndBox;
```

```
F_Periodo=Ok;
```


End;

Function F_NivelA(Cod)

Ok=0;

Box Cod => Ok;

1-2 => 1;

3-4 => 2;

5-6 => 3;

7-11 => 4;

EndBox;

F_NivelA = Ok;

End;

Function F_NivelB(Cod)

Box Cod => Ok;

1-4 => 1;

5-6 => 2;

7-11 => 3;

EndBox;

F_NivelB=Ok;

End;

Function F_NivelC(Cod)

Box Cod => Ok;

1-6 => 1;

7-11 => 2;

EndBox;

F_NivelC=Ok;

End;

PROC EA03401A

PreProc

Level 0

Parametr = ToNumber(SysParm());

Level 1

If par300 <> 2 Or par400 <> 2 Or par500 <> 2 Then

Skip Case;

Endif;

```
r=LoadCase( Hogares,pConglo,pCodViv,pHog1,pHog2 );
If r=0 Then
  Skip Case;
Else
  If HNIvel=Missing Or      HNIvel=NotAppl Or HNIvel=0 Then
    Skip Case;
  Endif;
Endif;
```

PostProc

Level 1:

If F_ConyugeDona() Then

Ed = 0;

```
If Ed=0 & Eda300 > 0      Then
  Ed=Eda300;
Endif;
```

```
If Ed=0 & Edaa400 > 0 Then
  Ed=Edaa400;
Endif;
```

```
If Ed=0 & Eda500 > 0      Then
  Ed=Eda500;
Endif;
```

Sexo = 0;

```
If Sexo=0 & Sex300 >      0 Then
  Sexo=Sex300;
Endif;
```

```
If Sexo=0 & Sex400 >      0 Then
  Sexo=Sex400;
Endif;
```

```
If Sexo=0 & Sex500 >      0 Then
  Sexo=Sex500;
Endif;
```

```
Mes    = ToNumber(      Mes_01A    );
Edad    = F_Edad( Ed );
Miembros = F_Miembros( HMiembros );
Periodo = F_Periodo( Mes );
```

```
If Parametr=1 Then
  Nivel = F_NivelA(      HNIvel );
Endif;
```

```
If Parametr=2 Then
  Nivel = F_NivelB(      HNNivel );
Endif;
```

```
If Parametr=3 Then
  Nivel = F_NivelC(      HNNivel );
Endif;
```

```
Export Format Record( pConglo, pCodViv, pHog1, pHog2, pPer, HEstrato, Edad,
Sexo, Miembros, Nivel, Periodo, HUbigeo, HZona, HManza, HSufmanz, HAer );
```

```
Endif;
```

Programa A003

Identifica y genera la relación del Resto Receptores.

Código

PROC A003

Function F_Resto()

```
Miembro = ( edaa400 < 3 & mie400=1 & par400 <> 8 & par400 <> 9 ) Or
  ( ( eda300 >= 3 & eda300 < 14 & mie300=1 & par300 <> 8 & par300 <>
    9 ) & ( edaa400 >= 3 & edaa400 < 14 & mie400=1 & par400 <> 8 & par400 <>
    9 ) ) Or
  ( ( eda300 >= 14 & eda300 <= 98 & mie300=1 & par300 <> 8 & par300 <>
    9 ) & ( edaa400 >= 14 & edaa400 <= 98 & mie400=1 & par400 <> 8 & par400 <> 9
    ) & ( eda500 >= 14 & eda500 <= 98 & mie500=1 & par500 <> 8 & par500 <> 9
    ) );
```

```
Omision = ( edaa400 < 3 & p400i = 0 ) Or
  ( ( eda300 >= 3 & eda300 < 14 & p300i = 0 ) & ( edaa400 >= 3 & edaa400
    <= 14 & p400i = 0 ) ) Or
  ( ( eda300 >= 14 & eda300 <= 98 & p300i = 0 ) & ( edaa400 >= 14 &
    edaa400 <= 98 & p400i = 0 ) & ( eda500 >= 14 & eda500 <= 98 & p500i = 0 ) );
```

```
Resto = ( par400 >= 3 ) Or
  ( par300 >= 3 & par400 >= 3 ) Or
  ( par300 >= 3 & par400 >= 3 & par500 >= 3 );
```

```
F_Resto = Miembro & Omision & Resto;
```

```
End;
```

Function F_Miembros(Cod)

```
Box Cod      => Ok;
```

```
1-2 => 1;
```

```
3-4 => 2;
```

```
5-6 => 3;
7-8 => 4;
>=9 => 5;
EndBox;

F_Miembros = Ok;

End;

Function F_Edad( Cod )

Ok=0;
Box Cod =>      Ok;
  0  =>  1;
 1-4 =>  2;
  5  =>  3;
  6  =>  4;
  7  =>  5;
  8  =>  6;
  9  =>  7;
 10  =>  8;
 11  =>  9;
 12  => 10;
 13  => 11;
 14  => 12;
 15  => 13;
 16  => 14;
 17  => 15;
 18  => 16;
 19  => 17;
 20  => 18;
 21  => 19;
 22  => 20;
 23  => 21;
 24  => 22;
25-50 => 23;
51-64 => 24;
65-98 => 25;
Endbox;

F_Edad=Ok;

End;

Function F_Periodo( Cod      )

Box Cod      => Ok;
  1-3  => 1;
  4-6  => 2;
  7-9  => 3;
```

10-12 => 4;

EndBox;

F_Periodo=Ok;

End;

Function F_NivelA(Cod)

Ok=0;

Box Cod => Ok;

1-2 => 1;

3-4 => 2;

5-6 => 3;

7-11 => 4;

EndBox;

F_NivelA=Ok;

End;

Function F_NivelB(Cod)

Box Cod => Ok;

1-4 => 1;

5-6 => 2;

7-11 => 3;

EndBox;

F_NivelB=Ok;

End;

Function F_NivelC(Cod)

Box Cod => Ok;

1-6 => 1;

7-11 => 2;

EndBox;

F_NivelC=Ok;

End;

PROC EA03401A

PreProc

Level 0

Parametr = ToNumber(SysParm());

Level 1

```
r=LoadCase( Hogares,pConglo,pCodViv,pHog1,pHog2 );
If r=0 Then
  Skip Case;
Else
  If HNIvel=Missing Or      HNIvel=0 Or HNIvel=NotAppl Then
    Skip Case;
  Endif;
Endif;
```

PostProc

Level 1

If F_Resto() Then

Ed=0;

```
If Ed=0 & Eda300 > 0      Then
  Ed=Eda300;
Endif;
```

```
If Ed=0 & Edaa400 > 0 Then
  Ed=Edaa400;
Endif;
```

```
If Ed=0 & Eda500 > 0      Then
  Ed=Eda500;
Endif;
```

Sexo = 0;

```
If Sexo=0 & Sex300 >      0 Then
  Sexo=Sex300;
Endif;
```

```
If Sexo=0 & Sex400 >      0 Then
  Sexo=Sex400;
Endif;
```

```
If Sexo=0 & Sex500 >      0 Then
  Sexo=Sex500;
Endif;
```

```
Edad   = F_Edad( Ed );
Periodo = F_Periodo( ToNumber( Mes_01A ) );
Miembros = F_Miembros( HMiembro );
```

```
If Parametr=1 Then
  Nivel = F_NivelA(      HNIvel );
Endif;
```

```
If Parametr=2 Then
  Nivel = F_NivelB(      HNivel );
Endif;
```

```
If Parametr=3 Then
  Nivel = F_NivelC( HNivel );
Endif;
```

```
Export Format Record( pConglo, pCodViv, pHog1, pHog2, pPer, HEstrato, Edad,
Sexo, Miembros, Nivel , Periodo,  HUbigeo, HZona,  HManza,  HSufManz,
HAer );
```

```
Endif;
```

Programa A008

Identifica y genera la relación del Resto Donantes.

Código

PROC A008

Function F_RestoDona()

```
Completo = ( edaa400 < 3 & p400i > 0 ) Or
  ( ( eda300 >= 3 & eda300 < 14 & p300i > 0 ) & ( edaa400 >= 3 &
edaa400 <= 14 & p400i > 0 ) ) Or
  ( ( eda300 >= 14 & eda300 <= 98 & p300i > 0 ) & ( edaa400 >= 14 &
edaa400 <= 98 & p400i > 0 ) & ( eda500 >= 14 & eda500 <= 98 & p500i > 0
) );
```

```
Resto = ( edaa400 < 3 & par400 >= 3 ) Or
  ( ( eda300 >= 3 & eda300 < 14 & par300 >= 3 ) & ( edaa400 >= 3 & edaa400 <
14 & par400 >= 3 ) ) Or
  ( ( eda300 >= 14 & eda300 <= 98 & par300 >= 3 ) & ( edaa400 >= 14 &
edaa400 <= 98 & par400 >= 3 ) & ( eda500 >= 14 & eda500 <= 98 & par500 >= 3 ) );
```

```
Miembro = ( edaa400 < 3 & mie400=1 & par400 <> 8 & par400 <> 9 ) Or
  ( ( eda300 >= 3 & eda300 < 14 & mie300=1 & par300 <> 8 & par300 <>
9 ) & ( edaa400 >= 3 & edaa400 < 14 & mie400=1 & par400 <> 8 & par400 <>
9 ) ) Or
  ( ( eda300 >= 14 & eda300 <= 98 & mie300=1 & par300 <> 8 & par300 <>
9 ) & ( edaa400 >= 14 & edaa400 <= 98 & mie400=1 & par400 <> 8 & par400 <> 9 )
& ( eda500 >= 14 & eda500 <= 98 & mie500=1 & par500 <> 8 & par500 <> 9 ) );
```

```
F_RestoDona = Completo & Resto & Miembro;
```

```
End;
```

```
Function F_Edad( Cod );
```

```
Ok=0;
Box Cod => Ok;
  0  => 1;
  1-4 => 2;
  5  => 3;
  6  => 4;
  7  => 5;
  8  => 6;
  9  => 7;
 10  => 8;
 11  => 9;
 12  => 10;
 13  => 11;
 14  => 12;
 15  => 13;
 16  => 14;
 17  => 15;
 18  => 16;
 19  => 17;
 20  => 18;
 21  => 19;
 22  => 20;
 23  => 21;
 24  => 22;
 25-50 => 23;
 51-64 => 24;
 65-98 => 25;
Endbox;
```

```
F_Edad=Ok;
```

```
End;
```

```
Function F_Miembros( Cod )
```

```
Ok=0;
Box Cod => Ok;
  1-2 => 1;
  3-4 => 2;
  5-6 => 3;
  7-8 => 4;
  >=9 => 5;
EndBox;
```

```
F_Miembros=Ok;
```

```
End;
```

```
Function F_Periodo( Cod )
```



```
Ok=0;
Box Cod => Ok;
  1-3  => 1;
  4-6  => 2;
  7-9  => 3;
 10-12 => 4;
EndBox;

F_Periodo=Ok;

End;
```

Function F_NivelA(Cod)

```
Ok=0;
Box Cod  => Ok;
  1-2  => 1;
  3-4  => 2;
  5-6  => 3;
  7-11 => 4;
EndBox;

F_NivelA=Ok;

End;
```

Function F_NivelB(Cod)

```
Box Cod      => Ok;
  1-4  => 1;
  5-6  => 2;
  7-11 => 3;
EndBox;

F_NivelB=Ok;

End;
```

Function F_NivelC(Cod)

```
Box Cod      => Ok;
  1-6  => 1;
  7-11 => 2;
EndBox;

F_NivelC=Ok;
```

```
End;
PROC EA03401A
PreProc
```

Level 0

Parametr = ToNumber(SysParm());

Level 1

r=LoadCase(Hogares,pConglo,pCodViv,pHog1,pHog2);

If r=0 Then

 Skip Case;

Else

 If HNNivel=Missing Or HNNivel=0 Or HNNivel=NotAppl Then

 Skip Case;

 Endif;

Endif;

PostProc

Level 1:

If F_RestoDona() Then

 Ed=0;

 If Ed=0 & Eda300 > 0 Then

 Ed=Eda300;

 Endif;

 If Ed=0 & Edaa400 > 0 Then

 Ed=Edaa400;

 Endif;

 If Ed=0 & Eda500 > 0 Then

 Ed=Eda500;

 Endif;

 Sexo=0;

 If Sexo=0 & Sex300 > 0 Then

 Sexo=Sex300;

 Endif;

 If Sexo=0 & Sex400 > 0 Then

 Sexo=Sex400;

 Endif;

 If Sexo=0 & Sex500 > 0 Then

 Sexo=Sex500;

 Endif;

 Mes = ToNumber(Mes_01A);

 Edad = F_Edad(Ed);

 Miembros = F_Miembros(HMiembro);

```
Periodo = F_Periodo( Mes );

If Parametr=1 Then
    Nivel = F_NivelA( HNivel );
Endif;

If Parametr=2 Then
    Nivel = F_NivelB( HNivel );
Endif;

If Parametr=3 Then
    Nivel = F_NivelC( HNivel );
Endif;

Export Format Record( pConglo, pCodViv, pHog1, pHog2, pPer, HEstrato, Edad,
Sexo, Miembros,Nivel,Periodo,HUbigeo,HZona,HManza,HSufmanz,HAer );

Endif;
```

Programa AsigDona

Este programa establece la relación única entre el receptor y donante para tres grandes grupos de parentesco : Jefes de Hogar, Cónyuges y Resto. Es decir, en términos prácticos genera el archivo correspondiente en donde figuran el Receptor junto con el Donante respectivo si lo hubiese para los : Jefes de Hogar, Cónyuges y Resto.

La asignación de donantes se realiza en base a 3 iteraciones :

1ra. Iteración : J_Estra , J_Edad , J_Sexo , J_Miem , J_Nivel , J_Peri

2da. Iteración : J_Estra, J_Edad, J_Sexo , J_Miem , J_Nivel

3ra. Iteración : J_Edad , J_Sexo , J_Miem , J_Nivel

Breve descripción del proceso.

El proceso toma el total de receptores y le aplica la 1ra. Iteración, de quedar receptores sin donantes son considerados para una 2da. Iteración, si en ésta hay receptores que se quedan sin donantes, éstos pasan a una 3ra. Iteración. Y como producto de esta 3ra. y última iteración quedan receptores sin donantes.

Código

Parameter Dpto,Anio,Modo

*

Do Case

*

Case Modo="1"

*

 Titulo = "Asignando Jefes de Hogar"

 ArchOmi = "OmiJ" + Dpto + "00.Dat"

 ArchDon = "DonJ" + Dpto + "00.Dat"

 ArchAsg = "DoJe" + Dpto + "00.Dat"

*

Case Modo="2"

*

 Titulo = "Asignando Conyuges"

 ArchOmi = "OmiE" + Dpto + "00.Dat"

 ArchDon = "DonE" + Dpto + "00.Dat"

 ArchAsg = "DoEs" + Dpto + "00.Dat"

*

Case Modo="3"

*

 Titulo = "Asignando Resto"

 ArchOmi = "OmiR" + Dpto + "00.Dat"

 ArchDon = "DonR" + Dpto + "00.Dat"

 ArchAsg = "DoRe" + Dpto + "00.Dat"

*

EndCase

*

SET TALK OFF

SET SCOREBOARD OFF

SET ESCAPE OFF

SET CONSOLE OFF

SET SAFETY OFF

SET DATE TO BRITISH

SET DELETE ON

SET COLOR TO W+/N

*

@ 00,00 CLEAR TO 24,79

@ 00,00 TO 24,79

@ 01,01 SAY padc('Encuesta Nacional de Hogares ' + Anio,78) COLOR W+/B

@ 11,01 Say padc(Titulo,78) Color W+/N

@ 12,01 Say padc("Procesando",78) Color W+*/N

*

Sele 0

Create Table OMISOS (J_CONGLO C(4), J_CODVIV C(3), J_HOGAR C(2),
J_PERSON C(2), J_ESTR A C(1), J_EDAD C(2), J_SEXO C(1), J_MIEM C(1),
J_NIVEL C(1), J_PERI C(1), J_UBIGEO C(6), J_ZONA C(3), J_MANZA C(3),
J_SUFMA C(1), J_AER C(3), D_CONGLO C(4), D_CODVIV C(3), D_HOGAR C(2),
D_PERSON C(2), Pasada N(1))

Index On

J_ESTR A+J_EDAD+J_SEXO+J_MIEM+J_NIVEL+J_PERI+J_UBIGEO+J_ZONA+J
_MANZA+J_SUFMA+J_AER+J_CONGLO+J_CODVIV+J_HOGAR+J_PERSON

Tag Key01

Index On

J_ESTR A+J_EDAD+J_SEXO+J_MIEM+J_NIVEL+J_UBIGEO+J_ZONA+J_MANZ
A+J_SUFMA+J_AER+J_CONGLO+J_CODVIV+J_HOGAR+J_PERSON Tag

Key01A

Index On

J_EDAD+J_SEXO+J_MIEM+J_NIVEL+J_UBIGEO+J_ZONA+J_MANZA+J_SUFM
A+J_AER+J_CONGLO+J_CODVIV+J_HOGAR+J_PERSON Tag Key01B

*

Copy Stru To TmpOmiso

Archivo = "Dat\" + ArchOmi

Append From (Archivo) sdf

*

Sele 0

Create Table DONANTES (J_CONGLO C(4), J_CODVIV C(3), J_HOGAR C(2),
J_PERSON C(2), J_ESTR A C(1), J_EDAD C(2), J_SEXO C(1), J_MIEM C(1),
J_NIVEL C(1), J_PERI C(1), J_UBIGEO C(6), J_ZONA C(3), J_MANZA C(3),
J_SUFMA C(1), J_AER C(3), J_ESTADO L)

Index On

J_ESTR A+J_EDAD+J_SEXO+J_MIEM+J_NIVEL+J_PERI+J_UBIGEO+J_ZONA+J
_MANZA+J_SUFMA+J_AER+J_CONGLO+J_CODVIV+J_HOGAR+J_PERSON

Tag Key01

Index On

J_ESTR A+J_EDAD+J_SEXO+J_MIEM+J_NIVEL+J_UBIGEO+J_ZONA+J_MANZ
A+J_SUFMA+J_AER+J_CONGLO+J_CODVIV+J_HOGAR+J_PERSON Tag

Key01A

```
Index On
J_EDAD+J_SEXO+J_MIEM+J_NIVEL+J_UBIGEO+J_ZONA+J_MANZA+J_SUFM
A+J_AER+J_CONGLO+J_CODVIV+J_HOGAR+J_PERSON Tag Key01B
Index On J_CONGLO+J_CODVIV+J_HOGAR+J_PERSON Tag Key02
*
Archivo = "Dat\" + ArchDon
Append From (Archivo) sdf
*
*----- Asigna Donantes -----*
*
Sele 0
Use TmpOmisos
Zap
Index On
J_ESTR+J_EDAD+J_SEXO+J_MIEM+J_NIVEL+J_PERI+J_UBIGEO+J_ZONA+J
_MANZA+J_SUFMA+J_AER+J_CONGLO+J_CODVIV+J_HOGAR+J_PERSON
Tag Key01
Index On
J_ESTR+J_EDAD+J_SEXO+J_MIEM+J_NIVEL+J_UBIGEO+J_ZONA+J_MANZ
A+J_SUFMA+J_AER+J_CONGLO+J_CODVIV+J_HOGAR+J_PERSON Tag
Key01A
Index On
J_EDAD+J_SEXO+J_MIEM+J_NIVEL+J_UBIGEO+J_ZONA+J_MANZA+J_SUFM
A+J_AER+J_CONGLO+J_CODVIV+J_HOGAR+J_PERSON Tag Key01B
*
Sele Omisos
*
Do Paso1
Do Paso2
Do Paso3
*
Archivo="Dat\" + ArchAsg
Sele 0
Select J_Conglo+J_CodViv+J_Hogar+J_Person As Clave, "1" as Id, J_CONGLO,
J_CODVIV, J_HOGAR, J_PERSON, J_ESTR, J_EDAD, J_SEXO, J_MIEM,
J_NIVEL, J_PERI, J_UBIGEO, J_ZONA, J_MANZA, J_SUFMA, J_AER,
D_CONGLO, D_CODVIV, D_HOGAR, D_PERSON From Omisos into table Tmp2
Use
*
Sele 0
Use Tmp2
Copy To (Archivo) sdf
*
Close All
*
Return

*-----*
Procedure Paso1
*-----*
```

```
Sele Omisos
Set Order To Tag Key01
Go Top
Do While !Eof()
  *
  If Empty( D_Conglo )
    *
    Replace Pasada With 1
    Scatter memvar
    Clave = m.J_Estra + m.J_Edad + m.j_sexo + m.j_miem + m.j_nivel + m.j_peri
    *
    Sele TmpOmiso
    Set Order To Tag Key01
    Zap
    *
    Sele Donantes
    Set Order To Tag Key01
    If Seek( Clave )
      Do While Clave = J_Estra + J_Edad + J_Sexo + J_Miem + J_Nivel + J_Peri And
!Eof()
        If !J_Estado
          Scatter MemVar
          Sele TmpOmiso
          Append Blank
          Gather MemVar
        Endif
        Sele Donantes
        Skip
      Enddo
    Endif
    *
    IdPerson = Space(11)
    *
    Sele TmpOmiso
    Go Top
    If !Eof()
      *
      IdPerson = J_Conglo + J_CodViv + J_Hogar + J_Person
      *
      Sele Donantes
      Set Order To Tag Key02
      If Seek( IdPerson )
        Replace J_Estado With .T.
      Endif
      *
      Sele Omisos
      Replace D_Conglo With TmpOmiso.J_Conglo
      Replace D_CodViv With TmpOmiso.J_CodViv
      Replace D_Hogar With TmpOmiso.J_Hogar
      Replace D_Person With TmpOmiso.J_Person
```

```
        Replace Pasada With TmpOmiso.Pasada
    *
Endif
*
Endif
*
Sele Omisos
Skip
*
Enddo
*
Return

*-----*
Function Paso2
*-----*
Sele Omisos
Set Order To Tag Key01A
Go Top
Do While !Eof()
    *
    If Empty( D_Conglo )
        *
        Replace Pasada With 2
        Scatter memvar
        Clave = J_Estra + J_Edad + J_Sexo + J_Miem + J_Nivel
        *
        Sele TmpOmiso
        Set Order To Tag Key01A
        Zap
        *
        Sele Donantes
        Set Order To Tag Key01A
        If Seek( Clave )
            Do While Clave = J_Estra + J_Edad + J_Sexo + J_Miem + J_Nivel And !Eof()
                If !J_Estado
                    Scatter MemVar
                    Sele TmpOmiso
                    Append Blank
                    Gather MemVar
                Endif
                Sele Donantes
                Skip
            Enddo
        Endif
        *
        IdPerson = Space(11)
        *
        Sele TmpOmiso
        Go Top
```



```
If !Eof()
*
  IdPerson = J_Conglo + J_CodViv + J_Hogar + J_Person
*
  Sele Donantes
  Set Order To Tag Key02
  If Seek( IdPerson )
    Replace J_Estado With .T.
  Endif
*
  Sele Omisos
  Replace D_Conglo With TmpOmiso.J_Conglo
  Replace D_CodViv With TmpOmiso.J_CodViv
  Replace D_Hogar With TmpOmiso.J_Hogar
  Replace D_Person With TmpOmiso.J_Person
  Replace Pasada With TmpOmiso.Pasada
*
  Endif
*
Endif
*
Sele Omisos
Skip
*
Enddo
*
Return

*-----*
Function Paso3
*-----*
Sele Omisos
Set Order To Tag Key01B
Go Top
Do While !Eof()
*
  If Empty( D_Conglo )
    *
    Replace Pasada With 3
    Scatter memvar
    Clave = J_Edad + J_Sexo + J_Miem + J_Nivel
    *
    Sele TmpOmiso
    Set Order To Tag Key01B
    Zap
    *
    Sele Donantes
    Set Order To Tag Key01B
    If Seek( Clave )
      Do While Clave = J_Edad + J_Sexo + J_Miem + J_Nivel And !Eof()
```

```

    If !J_Estado
        Scatter MemVar
        Sele TmpOmiso
        Append Blank
        Gather MemVar
    Endif
    Sele Donantes
    Skip
Enddo
Endif
*
IdPerson = Space(11)
*
Sele TmpOmiso
Go Top
If !Eof()
    *
    IdPerson = J_Conglo + J_CodViv + J_Hogar + J_Person
    *
    Sele Donantes
    Set Order To Tag Key02
    If Seek( IdPerson )
        Replace J_Estado With .T.
    Endif
    *
    Sele Omisos
    Replace D_Conglo With TmpOmiso.J_Conglo
    Replace D_CodViv With TmpOmiso.J_CodViv
    Replace D_Hogar With TmpOmiso.J_Hogar
    Replace D_Person With TmpOmiso.J_Person
    Replace Pasada With TmpOmiso.Pasada
    *
Endif
*
Endif
*
Sele Omisos
Skip
*
Enddo
*
Return
```

Programa A009

Realiza la imputación de los datos del donante en su receptor correspondiente. Esta relación se encuentra en los archivos generados por el programa AsigDona para los Jefes de Hogar, Cónyuges y el Resto.

El programa extrae los datos del donante, cambia la identificación del mismo por la identificación del receptor, con esto los datos pasan a ser del receptor.

Es de importancia mencionar que los datos de los donantes correspondientes a las variables cualitativas son mantenidos a diferencia de los datos para las variables cuantitativas (ingresos y gastos) los cuales son cambiados a “missing”. Posteriormente los procesos de imputación considerados dentro del procesamiento de datos tradicional, serán los encargados de asignar a estos valores “missing” los valores promedios correspondientes si los hubiese.

Código

PROC A009

Function Limpiar()

z=1;

n300 = noccurs(cap300);

While z <= n300 Do;

 If p311b(z) > 0 Then

 p311b(z) = Missing;

 d311b(z) = Missing;

 i311b(z) = NotAppl;

 Endif;

 If p311d(z) > 0 Then

 p311d(z) = Missing;

 d311d(z) = Missing;

 i311d(z) = NotAppl;

 Endif;

 z=z+1;

Enddo;

If P311T1 > 0 Then

 P311T1 = Missing;

Endif;

If P311T2 > 0 Then

 P311T2 = Missing;

Endif;

If P3121B > 0 Then

 P3121B = Missing;

 D3121B = Missing;

 I3121B = NotAppl;

Endif;

If P3121C > 0 Then

 P3121C = Missing;

 D3121C = Missing;

 I3121C = NotAppl;

Endif;

If P3122B > 0 Then
 P3122B = Missing;
 D3122B = Missing;
 I3122B = NotAppl;
Endif;

If P3122C > 0 Then
 P3122C = Missing;
 D3122C = Missing;
 I3122C = NotAppl;
Endif;

If P312T1 > 0 Then
 P312T1 = Missing;
Endif;

If P312T2 > 0 Then
 P312T2 = Missing;
Endif;

If P314B > 0 Then
 P314B = Missing;
 D314B = Missing;
 I314B = NotAppl;
Endif;

If P314C > 0 Then
 P314C = Missing;
 D314C = Missing;
 I314C = NotAppl;
Endif;

z=1;
n400 = noccurs(Cap400);
While z <= n400 Do;

 If p416(z) > 0 Then
 p416(z) = Missing;
 d416(z) = Missing;
 I416(z) = NotAppl;
 Endif;

 If p418(z) > 0 Then
 p418(z) = Missing;
 d418(z) = Missing;
 i418(z) = NotAppl;
 Endif;

z=z+1;

Enddo;

If P513 = 0 Then

P513 = Missing;

P513A = Missing;
I513A = NotAppl;

P513B = Missing;
I513B = NotAppl;

P513C = Missing;
I513C = NotAppl;

P513D = Missing;
I513D = NotAppl;

P513E = Missing;
I513E = NotAppl;

P513F = Missing;
I513F = NotAppl;

P513G = Missing;
I513G = NotAppl;

P513T = Missing;
I513T = NotAppl;

Endif;

If P518 > 0 Then

P518 = Missing;
I518 = NotAppl;

Endif;

If P520 > 0 Then

P520 = Missing;
I520 = NotAppl;

Endif;

If P524A1 > 0 Then

P524A1 = Missing;
D524A1 = Missing;
I524A1 = NotAppl;
P524A2 = 1;

Endif;

```
If P524B1 > 0 Then  
    P524B1 = Missing;  
    D524B1 = Missing;  
    I524B1 = NotAppl;  
    P524B2 = 1;  
Endif;
```

```
If P524B1 > 0 Then  
    P524CC1 = Missing;  
    D524CC1 = Missing;  
    I524CC1 = NotAppl;  
    P524CC2 = 1;  
Endif;
```

```
If P524C1 > 0 Then  
    P524C1 = Missing;  
    D524C1 = Missing;  
    I524C1 = NotAppl;  
    P524C2 = 1;  
Endif;
```

```
If P524D1 > 0 Then  
    P524D1 = Missing;  
    D524D1 = Missing;  
    I524D1 = NotAppl;  
    P524D2 = 1;  
Endif;
```

```
If P530A > 0 Then  
    P530A = Missing;  
    D530A = Missing;  
    I530A = NotAppl;  
Endif;
```

```
If P536 > 0 Then  
    P536 = Missing;  
    D536 = Missing;  
    I536 = NotAppl;  
Endif;
```

```
If P5291B > 0 Then  
    P5291B = Missing;  
Endif;
```

```
If P5292B > 0 Then  
    P5292B = Missing;  
Endif;
```

```
If P5293B > 0 Then
```

P5293B = Missing;
Endif;

If P5294B > 0 Then
P5294B = Missing;
D5294B = Missing;
I5294B = NotAppl;
Endif;

If P5295B > 0 Then
P5295B = Missing;
Endif;

If P5296B > 0 Then
P5296B = Missing;
Endif;

If P529T > 0 Then
P529T = Missing;
D529T = Missing;
Endif;

If P530A > 0 Then
P530A = Missing;
D530A = Missing;
I530A = NotAppl;
Endif;

If P536 > 0 Then
P536 = Missing;
D536 = Missing;
I536 = NotAppl;
Endif;

If P538A1 > 0 Then
P538A1 = Missing;
D538A1 = Missing;
I538A1 = NotAppl;
P538A2 = 1;
Endif;

If P538B1 > 0 Then
P538B1 = Missing;
D538B1 = Missing;
I538B1 = NotAppl;
P538B2 = 1;
Endif;

If P538CC1 > 0 Then
P538CC1 = Missing;

```
D538CC1 = Missing;  
I538CC1 = NotAppl;  
P538CC2 = 1;  
Endif;
```

```
If P538C1 > 0 Then  
    P538C1 = Missing;  
    D538C1 = Missing;  
    I538C1 = NotAppl;  
    P538C2 = 1;  
Endif;
```

```
If P538D1 > 0 Then  
    P538D1 = Missing;  
    D538D1 = Missing;  
    I538D1 = NotAppl;  
    P538D2 = 1;  
Endif;
```

```
If P5401B > 0 Then  
    P5401B = Missing;  
Endif;
```

```
If P5402B > 0 Then  
    P5402B = Missing;  
Endif;
```

```
If P5403B > 0 Then  
    P5403B = Missing;  
Endif;
```

```
If P5404B > 0 Then  
    P5404B = Missing;  
    D5404B = Missing;  
    I5404B = NotAppl;  
Endif;
```

```
If P5405B > 0 Then  
    P5405B = Missing;  
Endif;
```

```
If P5406B > 0 Then  
    P5406B = Missing;  
Endif;
```

```
If D540B > 0 Then  
    D540B = Missing;  
    I540B = NotAppl;  
Endif;
```



```
If D540T > 0 Then  
    D540T = Missing;  
Endif;
```

```
If P541A > 0 Then  
    P541A = Missing;  
    D541A = Missing;  
    I541A = NotAppl;  
Endif;
```

```
If P543 > 0 Then  
    P543 = Missing;  
Endif;
```

```
If D543 > 0 Then  
    D543 = Missing;  
    I543 = NotAppl;  
Endif;
```

```
If P5441B > 0 Then  
    P5441B = Missing;  
Endif;
```

```
If P5442B > 0 Then  
    P5442B = Missing;  
Endif;
```

```
If P5443B > 0 Then  
    P5443B = Missing;  
Endif;
```

```
If P5444B > 0 Then  
    P5444B = Missing;  
Endif;
```

```
If P5445B > 0 Then  
    P5445B = Missing;  
Endif;
```

```
If P5446B > 0 Then  
    P5446B = Missing;  
Endif;
```

```
If P5447B > 0 Then  
    P5447B = Missing;  
Endif;
```

```
If P5448B > 0 Then  
    P5448B = Missing;  
Endif;
```

If D544T > 0 Then
 D544T = Missing;
Endif;

If P5561C > 0 Then
 P5561C = Missing;
Endif;

If P5562C > 0 Then
 P5562C = Missing;
Endif;

If P5563C > 0 Then
 P5563C = Missing;
Endif;

If P5564C > 0 Then
 P5564C = Missing;
Endif;

If P5565C > 0 Then
 P5565C = Missing;
Endif;

If P5566C > 0 Then
 P5566C = Missing;
Endif;

If P5567C > 0 Then
 P5567C = Missing;
Endif;

If P5561E > 0 Then
 P5561E = Missing;
Endif;

If P5562E > 0 Then
 P5562E = Missing;
Endif;

If P5563E > 0 Then
 P5563E = Missing;
Endif;

If P5564E > 0 Then
 P5564E = Missing;
Endif;

If P5565E > 0 Then

P5565E = Missing;
Endif;

If P5566E > 0 Then
P5566E = Missing;
Endif;

If P5567E > 0 Then
P5567E = Missing;
Endif;

If P556T1 > 0 Then
P556T1 = Missing;
Endif;

If P556T2 > 0 Then
P556T2 = Missing;
Endif;

If D556T1 > 0 Then
D556T1 = Missing;
Endif;

If D556T2 > 0 Then
D556T2 = Missing;
Endif;

If P5571C > 0 Then
P5571C = Missing;
Endif;

If P5572C > 0 Then
P5572C = Missing;
Endif;

If P5573C > 0 Then
P5573C = Missing;
Endif;

If P5574C > 0 Then
P5574C = Missing;
Endif;

If P5575C > 0 Then
P5575C = Missing;
Endif;

If P5576C > 0 Then
P5576C = Missing;
Endif;

If P5577C > 0 Then
 P5577C = Missing;
Endif;

If P5578C > 0 Then
 P5578C = Missing;
Endif;

If P557T > 0 Then
 P557T = Missing;
Endif;

If D557T > 0 Then
 D557T = Missing;
Endif;

If P5581B > 0 Then
 P5581B = Missing;
Endif;

If P5582B > 0 Then
 P5582B = Missing;
Endif;

If P5583B > 0 Then
 P5583B = Missing;
Endif;

If P5584B > 0 Then
 P5584B = Missing;
Endif;

If P5585B > 0 Then
 P5585B = Missing;
Endif;

If P5586B > 0 Then
 P5586B = Missing;
Endif;

If P5587B > 0 Then
 P5587B = Missing;
Endif;

If P558T > 0 Then
 P558T = Missing;
Endif;

If D558T > 0 Then

```
D558T = Missing;
Endif;

z=1;
n500=noccurs( Cap500 );
While z <= n500 Do;

    If P559D(z) > 0 Then
        P559D(z) = Missing;
    Endif;

    z=z+1;

Enddo;

If D559D1 > 0 Then
    D559D1 = Missing;
    I559D1 = NotAppl;
Endif;

If D559D2 > 0 Then
    D559D2 = Missing;
    I559D2 = NotAppl;
Endif;

If D559D3 > 0 Then
    D559D3 = Missing;
    I559D3 = NotAppl;
Endif;

If D559D41 > 0 Then
    D559D41 = Missing;
    I559D41 = NotAppl;
Endif;

If D559D42 > 0 Then
    D559D42 = Missing;
    I559D42 = NotAppl;
Endif;

If D559D43 > 0 Then
    D559D43 = Missing;
    I559D43 = NotAppl;
Endif;

If D559D44 > 0 Then
    D559D44 = Missing;
    I559D44 = NotAppl;
Endif;
```

```
If D559D45 > 0 Then  
    D559D45 = Missing;  
    I559D45 = NotAppl;  
Endif;
```

```
If D559D46 > 0 Then  
    D559D46 = Missing;  
    I559D46 = NotAppl;  
Endif;
```

```
If P560D1 > 0 Then  
    P560D1 = Missing;  
    D560D1 = Missing;  
    I560D1 = NotAppl;  
Endif;
```

```
If P560D2 > 0 Then  
    P560D2 = Missing;  
    D560D2 = Missing;  
    I560D2 = NotAppl;  
Endif;
```

```
If P560D3 > 0 Then  
    P560D3 = Missing;  
    D560D3 = Missing;  
    I560D3 = NotAppl;  
Endif;
```

```
End;  
PROC DONANTES  
PostProc
```

Level 1

```
r=LoadCase( EA03401A,D_Conglo,D_CodViv,D_Hogar,D_Person    );  
If r=1 Then
```

```
    pConglo = j_Conglo;  
    pCodViv = j_Codviv;  
    pHog1 = j_Hogar/10;  
    pHog2 = j_Hogar%10;  
    pPer = j_Person;
```

```
    Persona=NotAppl;  
    r=LoadCase( EA034001,pConglo,pCodViv,pHog1,pHog2,Persona );  
    If r=1 Then
```

```
        PESTRA = Estrat01;  
        PDOMI = Domini01;  
        PUBIGEO = Ubigeo01;
```

ANIO_01A = AnioPro;

MES_01A = MesPro;

If p208a(pper) < 3 Then

 r = Clear(Cap300);

 r = Clear(Imput300);

Else

 p300n = pper;

 p300i = pper;

 par300 = p203(pper);

 sex300 = p207(pper);

 eda300 = p208a(pper);

 civ300 = p209(pper);

 mie300 = p204(pper);

 aus300 = p205(pper);

 pre300 = p206(pper);

Endif;

p400n = pper;

p400i = pper;

par400 = p203(pper);

sex400 = p207(pper);

edaa400 = p208a(pper);

edab400 = p208b(pper);

civ400 = p209(pper);

mie400 = p204(pper);

aus400 = p205(pper);

pre400 = p206(pper);

If p208a(pper) <= 13 Then

 r = Clear(Cap500);

 r = Clear(Imput500);

Else

 p500n = pper;

 p500i = pper;

 par500 = p203(pper);

 sex500 = p207(pper);

 eda500 = p208a(pper);

 civ500 = p209(pper);

 mie500 = p204(pper);

```
        aus500      = p205(pper);  
        pre500      = p206(pper);  
  
    Endif;  
  
Endif;  
  
r=Limpiar();  
  
r=WriteCase( EA03401A,pConglo,pCodViv,pHog1,pHog2,pPer );  
  
Endif;
```