



## TRABAJO PRÁCTICO N° 2

### Introducción

El presente trabajo representa un escenario típico de migración de datos desde un sistema preexistente hacia uno nuevo con estructuras de base de datos totalmente distintas. La información disponible inicialmente debe ser convertida, adecuada y depurada de acuerdo a la estructura propuesta para lo cual deben generarse elementos de información nuevos obtenidos y deducidos a partir de los datos iniciales y de los mecanismos necesarios que se planificaron para el nuevo sistema. Se considera solamente una parte del dominio del problema a los fines de la realización del práctico.

El sistema preexistente está desarrollado en Delphi en un ambiente cliente servidor y utiliza como servidor de base de datos SQLServer 2005. El sistema nuevo se desarrolla en Java dividiendo la lógica de la aplicación en capas (cliente, aplicaciones y base de datos). Se utilizará como servidor de base de datos SQLServer 20xx. Para el trabajo se proveen los siguientes elementos:

- Backup de la base de datos de SQLServer.
- Modelo de clases del sistema nuevo.

### El modelo preexistente

El sistema del trabajo consiste en la gestión de **Afiliaciones** de una entidad civil y la información que se provee en el backup mencionado solamente cuenta con tres tablas, **cuentas**, **copost** y **datosfam**. La tabla **cuentas** tiene toda la información de las personas afiliadas con la siguiente estructura:

Columna	Significado
<i>nrocta</i>	Número de identificación del afiliado.
<i>Tipdoc</i>	Tipo de documento del afiliado (1 es LE, 2 es LC, 3 es DNI y otro valor se considera como NO INFORMADO)
<i>nrodoc</i>	Número de documento del afiliado.
<i>nyap</i>	Apellido y nombre de afiliado.
<i>domicilio</i>	Domicilio <u>LEGAL</u> del afiliado.
<i>cp</i>	Código postal que determina la localidad del domicilio del afiliado.
<i>telefono</i>	Teléfono fijo del afiliado.
<i>celular</i>	Teléfono móvil del afiliado.

Columna	Significado
<i>nacionalidad</i>	Nacionalidad del afiliado.
<i>estacivil</i>	Estado civil actual del afiliado.
<i>fechanac</i>	Fecha de nacimiento del afiliado.
<i>sexo</i>	Género del afiliado.
<i>fecafiaa</i>	Fecha de alta del afiliado.
<i>fecafiba</i>	Fecha de baja del afiliado.

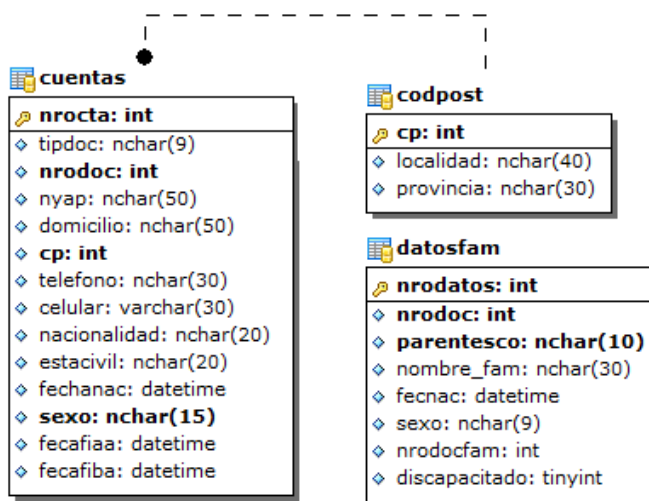
La tabla **codpost** tiene las localidades que se utilizan en el sistema con la siguiente estructura:

Columna	Significado
<i>cp</i>	Código postal de la localidad.
<i>localidad</i>	Nombre de la localidad.
<i>provincia</i>	Nombre de la provincia de la localidad.

La tabla **datosfam** tiene información de personas que son familiares de los afiliados indicando el vínculo de parentesco correspondiente. Tiene la siguiente estructura:

Columna	Significado
<i>nrodatos</i>	Número identificativo propio de esta tabla.
<i>nrodoc</i>	Número de documento del afiliado que tiene esta persona como familiar.
<i>parentesco</i>	Parentesco entre esta persona y el afiliado.
<i>nombre_fam</i>	Apellido y nombre de la persona.
<i>fecnac</i>	Fecha de nacimiento de la persona.
<i>sexo</i>	Género de la persona.
<i>nrodocfam</i>	Número de documento de la persona.
<i>discapacitado</i>	Si es discapacitado o no.

En esta última, solamente se dispone de número de documento (**nrodocfam**) para la identificación de la persona y para su relación con un afiliado se utiliza el número de documento de éste (**nrodoc**) – se asumirá que el tipo de documento es DNI.



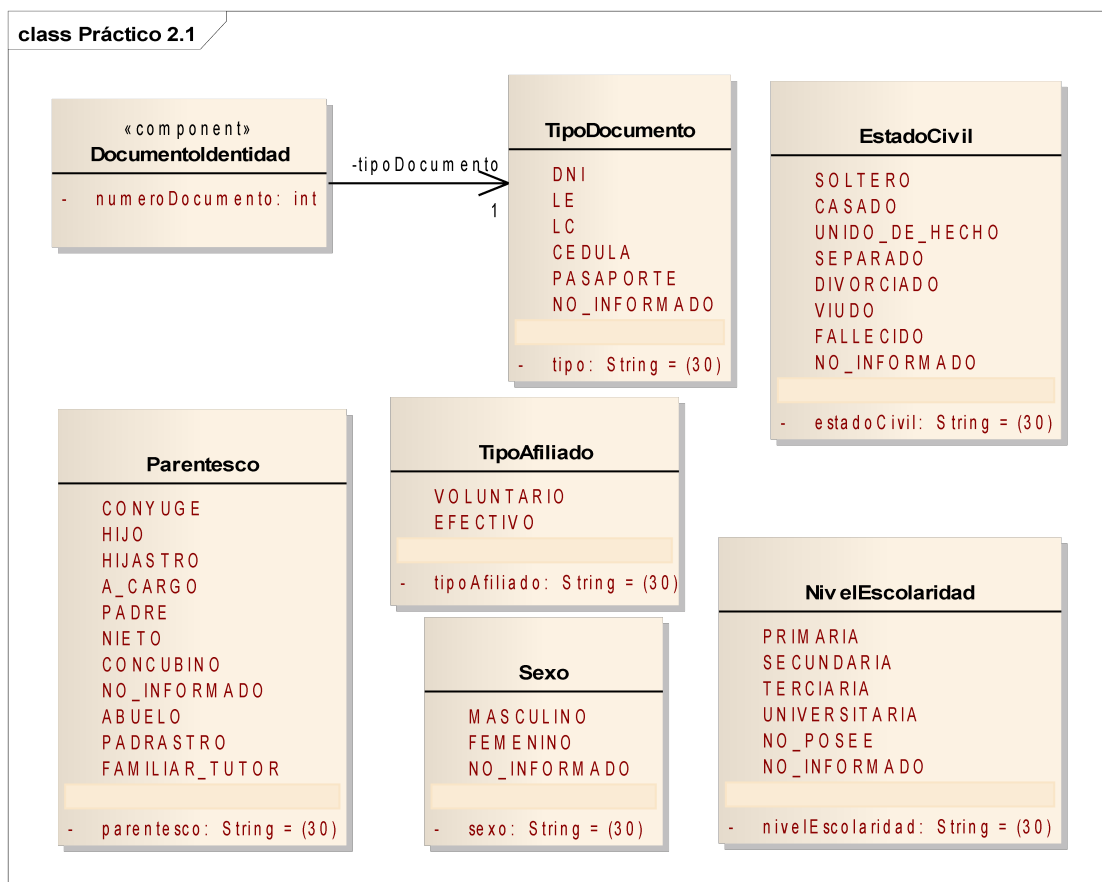
La integridad existente sólo es explícita entre **cuentas** y **codpost** mediante el join de la columna **cp** (código postal).

La tabla **cuentas** tiene información de cada recibo de sueldo presentado (aceptando solamente uno) y demás elementos de información no presentes en el práctico. El problema principal del sistema viejo era precisamente que al no existir el concepto de afiliado, al quererse registrar un nuevo detalle para otro recibo de sueldo, necesariamente se debía a cargar la misma persona otra vez como afiliado, o sea que una persona puede estar varias veces en esta tabla (tantas como cuentas distintas haya tenido incluso las que están dadas de baja). Las cuentas dadas de baja (que también se deben migrar) se distinguen porque la fecha de baja tiene valor.

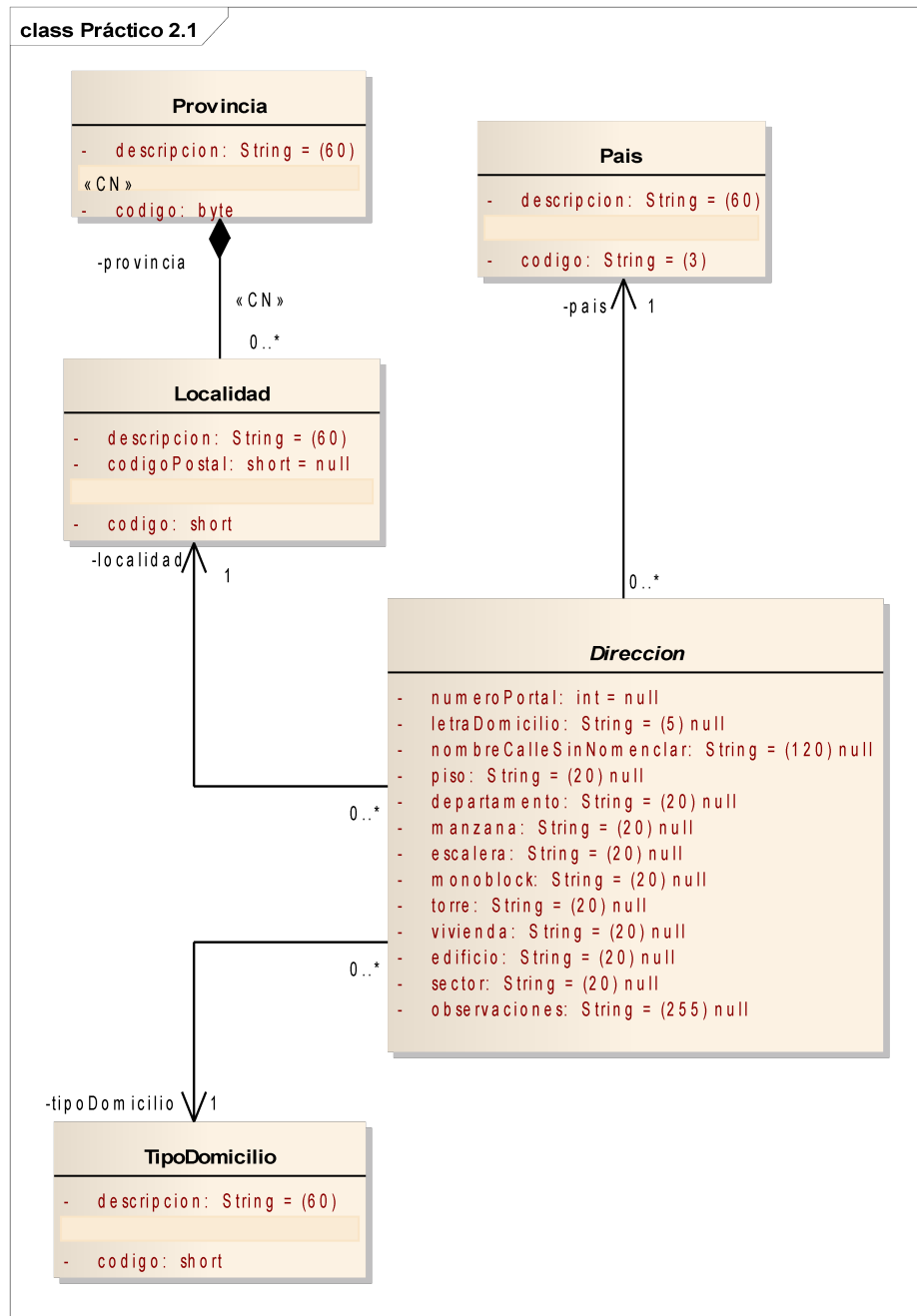
## Modelo destino

Como se indicara, la persistencia de la información del sistema nuevo se mantendrá también en un SQL Server, de manera que para realizar la migración se debe crear el modelo físico correspondiente que alojará la información requerida. Se presentan los siguientes modelos de clases a partir de los cuales se definirá la estructura en la base de datos. En el modelo se informa por cada atributo si este acepta o no nulo y además, si se trata de un String, la dimensión del mismo.

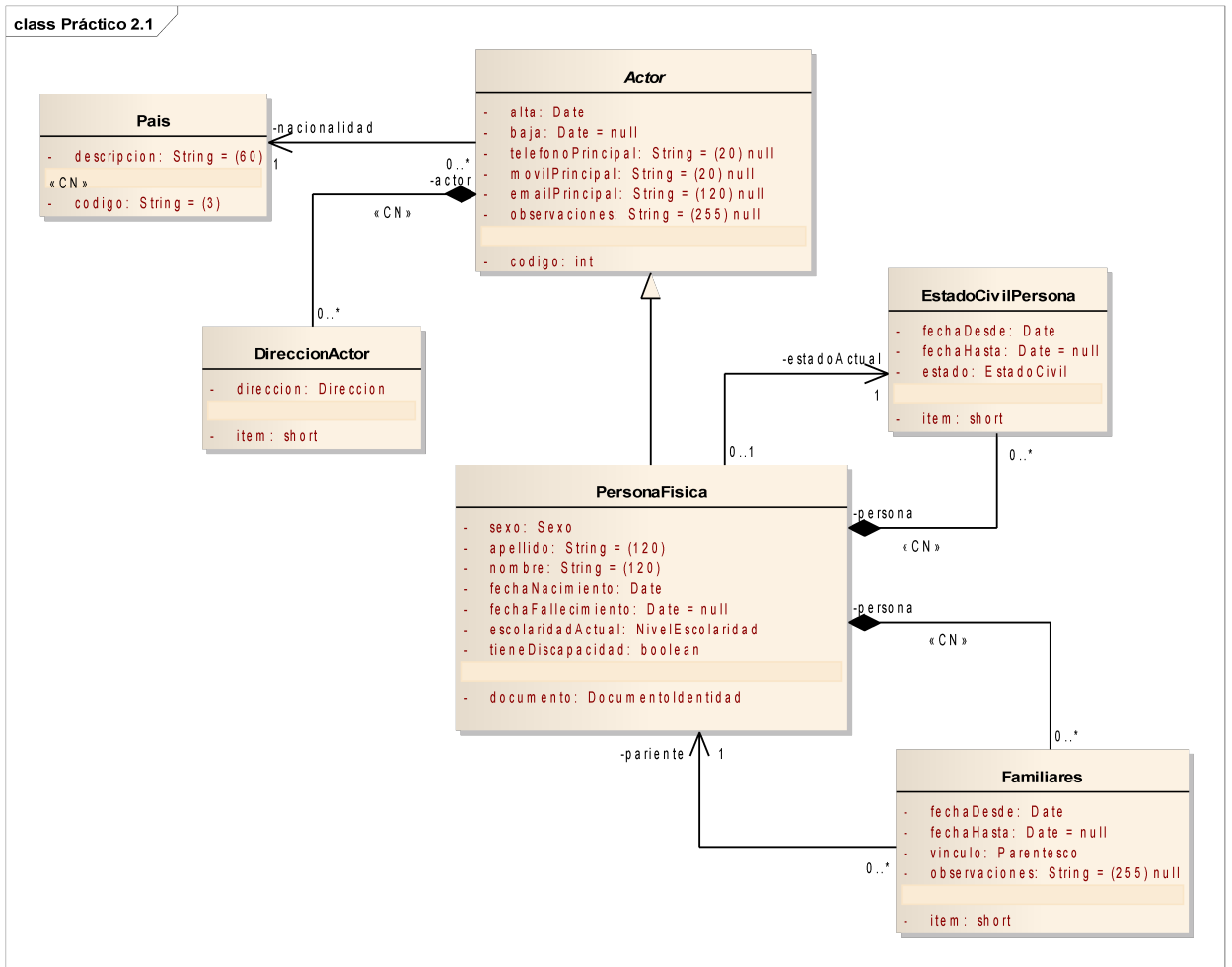
### 1. Clases no persistentes:



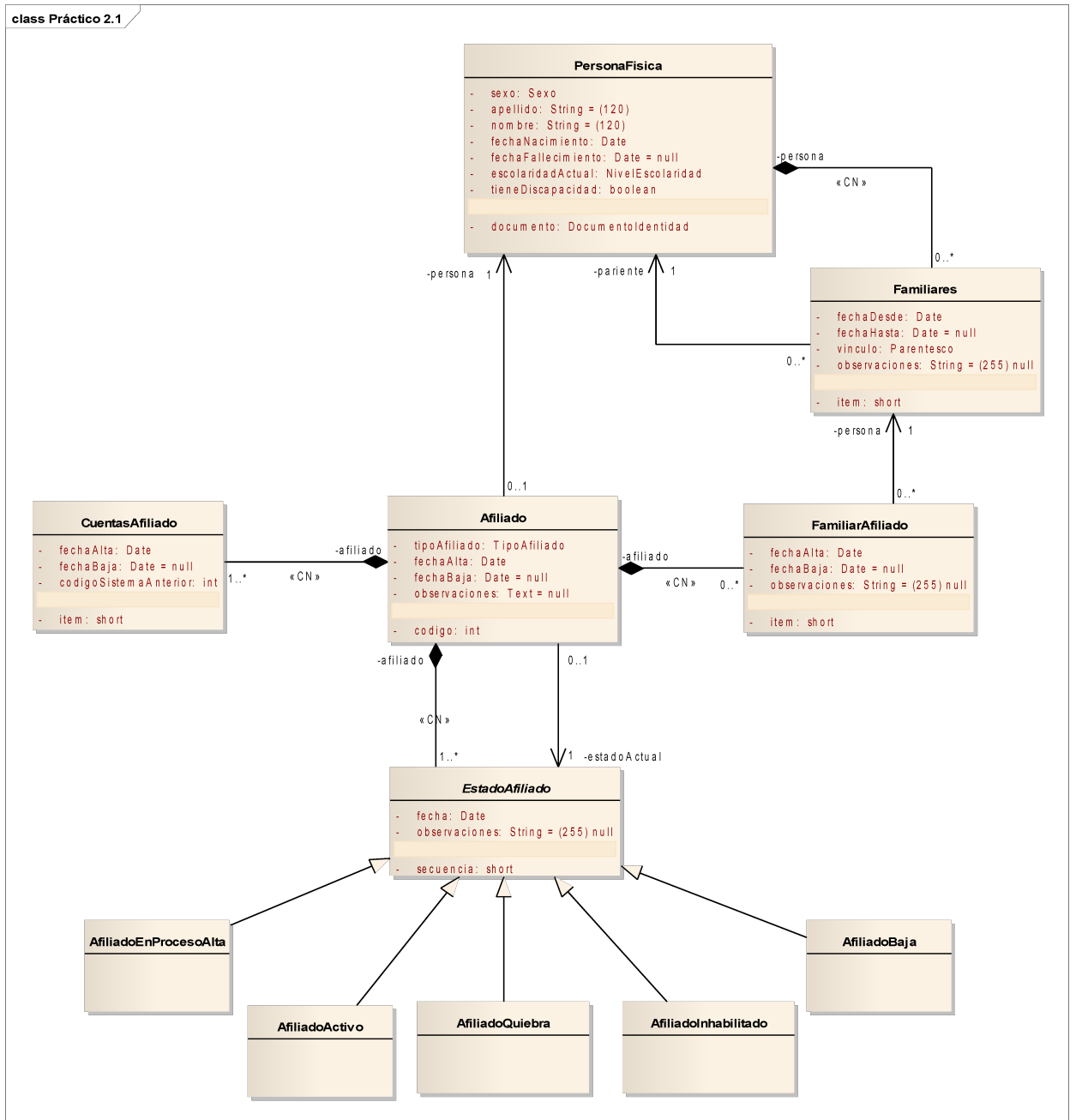
- Algunos de estos elementos se presentarán como *checks* en las tablas de la base.
2. Clases de clasificadores geográficos:



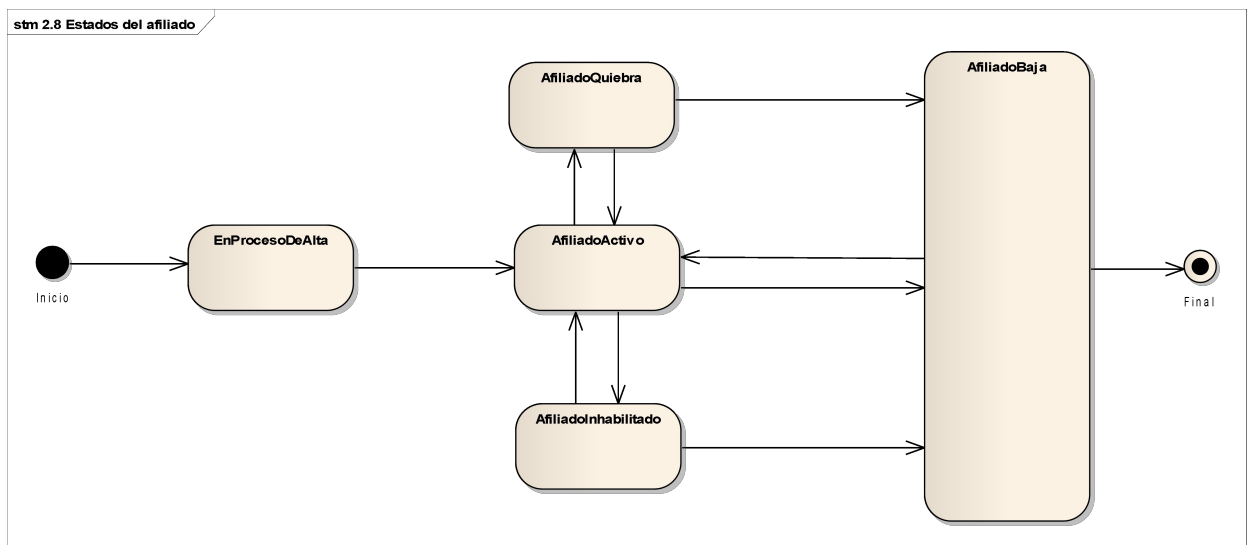
### 3. Clases del sistema de personas:



#### 4. Clases del sistema de afiliados:



Siendo la transición de estados para el afiliado:



### Aclaraciones sobre el modelo y la migración

La base de datos a crear tendrá como nombre la concatenación de los apellidos de los integrantes del grupo (separados por guión bajo). Las tablas de componentes geográficos y de personas, irán en el esquema **personas** y las de los afiliados en el esquema **afiliados**. Los nombres tanto de las tablas como de los atributos que representen palabras compuestas, se construirán con éstas palabras separadas por guión bajo (por ejemplo **nombreCalleSinNomenciar** será **nombre\_calle\_sin\_nomenclar**).

El modelo físico se debe generar con identificativos secuenciales (id) como primary key e implementar las claves de negocio como UNIQUE cuando corresponda.

Existen tablas que necesariamente se deben precargar u obtener de manera externa, tales como los países (clasificador internacional) y las provincias (generada a partir de los datos provistos en la migración). La tabla que corresponde al tipo de domicilio también debe precargarse con datos propios (del tipo de domicilio que se posee hasta el momento: LEGAL).

Existen tablas en que se deben interpretar los datos para hacer la conversión correspondiente, para ello se presenta el siguiente detalle acerca de cómo realizar el mapeo correspondiente:

#### Personas

- La fecha de alta (**alta**) del actor será el primer día del mes y año de la fecha de alta de la cuenta más vieja del afiliado (para el caso que la persona sea afiliado)
- La fecha de alta del actor que se tome de los familiares, será la misma que la del actor que es afiliado al que está relacionada. En caso de no existir, se tomará la fecha actual.
- La nacionalidad del actor deberá quedar reflejada en la relación con el país correspondiente.

- La dirección de la persona solamente se dispone como string de caracteres, razón por la cual tal información se registrará en la columna **nombre\_calle\_sin\_nomenclar** dejándose disponible para una depuración posterior (excede el presente práctico).
- El sexo (**sexo**) de la persona deberá tener el valor correspondiente al conjunto enumerado presentado por lo que deben realizarse las conversiones de los valores originales, en caso de presentarse un valor distinto a los previstos, se registrará como NO\_INFORMADO. La misma situación se presenta con los tipos de documento (**tipo\_documento**) y los estados civiles (**estado**); en este caso, si bien los estados civiles de la estructura actual permite mantener el historial, solamente se dispone de la situación actual (recordar mantener la referencia a éste). Ídem para los vínculos familiares (**vinculo**) que deberán mapearse convenientemente e indicarse NO\_INFORMADO para aquellos en que no pueda deducirse o encontrar correspondencia.
- El nivel de escolaridad (**escolaridad\_actual**) no está disponible en los datos anteriores por lo que se asignará NO\_INFORMADO en todos los casos.
- La discapacidad (**tiene\_discapacidad**) de la persona se asume como FALSE excepto que se informe lo contrario (dato existente en la tabla de familiares).
- Se debe separar el apellido del nombre de la persona.
- La fecha de inicio del familiar (**fecha\_desde**) será la fecha de nacimiento para el caso de HIJO, HIJASTRO, NIETO o la fecha de alta de la persona afiliada para los demás casos. El **ítem** es una secuencia que se inicia en 1 para cada persona de referencia y se incrementa en una unidad por cada familiar que tenga considerándose el orden cronológico (**fecha\_desde**).

### Afiliados

- Los tipos de afiliado (**tipo\_afiliado**) a migrar pertenecen en su totalidad al tipo EFECTIVO.
- El código (**codigo**) de afiliado es un número secuencial que se deberá generar.
- La fecha de alta (**fecha\_alta**) del afiliado será la menor de las fechas de alta de las cuentas asociadas.
- Si todas las cuentas asociadas (ahora en el detalle) tienen fecha de baja (**fecha\_baja**) no nula, se asume que el afiliado está dado de baja, siendo la fecha de baja del afiliado (**fecha\_baja**) la del último de los detalles. Dicha fecha además se la asigna al estado correspondiente (al estado BAJA).
- Los estados de los afiliados deben estar completos en la transición planificada, o sea que un afiliado de BAJA debe pasar necesariamente por EN\_PROCESO\_DE\_ALTA, ACTIVO y BAJA (que son los únicos que se presentarán de acuerdo a los datos iniciales). Para el caso, se asignará como fecha en el estado EN\_PROCESO\_DE\_ALTA la fecha de alta del afiliado, la del estado ACTIVO también la de la fecha de alta y la de BAJA la indicada en el punto anterior. Recordar de mantener la referencia al estado actual.
- La fecha de alta y de baja (**fecha\_alta**, **fecha\_baja**) del detalle de las cuentas del afiliado se obtienen directamente. Además **codigo\_sistema\_anterior** se corresponde al número de cuenta de los datos originales. El **ítem** es una secuencia que se inicia en 1 para cada afiliado y se incrementa en una unidad por cada cuenta que tenga considerándose el orden cronológico (**fecha\_alta**).
- Los familiares del afiliado tendrán como **fecha\_alta** la fecha de alta del afiliado. El **ítem** es similar al del punto anterior ya que inicialmente se considerará que todos los



familiares de la persona se registrarán también como familiares vinculados al afiliado correspondiente.

### Estrategia

Se recomienda el siguiente orden:



- Trabajar la parte geográfica. A partir de la tabla **codpost** se deben generar las distintas provincias y las localidades asociadas. Completar además las tabla de países con datos obtenidos de un nomenclador internacional. Completar la tabla de tipo de domicilio mínimamente con la ocurrencia de **LEGAL** que será referenciada luego.
- Identificar las distintas personas de las **cuentas** y migrar la más nueva (de mayor fecha de alta) porque se asumirá que tiene los datos más actualizados.
- Identificar los familiares y migrar haciendo simultáneamente el vínculo entre ellos (tabla **cuentas** y **datosfam**).
- Crear los afiliados.
- Crear el detalle de las cuentas de afiliado.
- Vincular afiliados con familiares.
- Crear los estados de los afiliados.
- Generar un log con los datos que no se han podido migrar indicando el motivo.

### **Consultas**

Construir las siguientes consultas:

1. Determinar el afiliado (**codigo**) que tiene mayor número de cuentas de detalle y que no estén dadas de baja.
2. Listado de afiliados que tienen como familiar a otro afiliado.
3. Dada una localidad, obtener la lista de afiliados que tienen domicilio en ella indicándose:
  - a. Tipo y número de documento
  - b. Apellido y nombre
  - c. Estado civil actual
  - d. Cantidad total de familiares vinculados
  - e. Cantidad total de cuentas que posee (activas o no).

### **Trigger**

Construir un trigger sobre la tabla correspondiente a los estados del afiliado de manera que solamente se permitan cambios válidos consecuentes con el diagrama de transición presentado.

## REALIZAR

El trabajo se debe estructurar en cuatro partes presentando cada una en archivos separados.

1. **Generación de la estructura de datos.** Se deberá construir el modelo físico (con cualquier herramienta) y entregar éste más el SQL de generación de las tablas, integridad e índices del nuevo sistema.
2. **Migración de datos.** se deberá construir un archivo con un script SQL que contenga los comandos que permitan la incorporación y acomodamiento de los datos en la base destino.
3. **Consultas.** Se deberá construir un archivo SQL que contenga los comandos select necesarios para responder las consultas indicadas.
4. **Trigger.** Se deberá construir un archivo SQL que contenga el código del trigger listo para aplicarse sobre la tabla correspondiente y presentar 10 comandos inserts, updates y deletes que prueben su funcionamiento..

Los scripts entregados deben ejecutarse sin errores en cualquier analizador de consultas de SQL Server (tipo SQL Server Management Studio o EMS Management).

**Vencimiento de la presentación: LUNES 17 DE NOVIEMBRE DE 2014**