

SGME - It5 - Análisis y Diseño de la Base de Datos

Requerimientos y definición de las tablas, columnas, dominios y restricciones necesarias para la persistencia de los datos del sistema.

Andrés Moschini

Historial de revisiones

Revisión Entrega007.01 2007-09-02

Creación del documento.

Revisión Entrega007.02 2007-09-09

Agregados procedimientos para obtener estados.

Revisión Entrega008.01 2007-09-23

Corrección de procedimientos para obtener los estados.

Revisión Entrega008.02 2007-09-24

Agregadas restricciones y modificación diagrama.

Revisión Entrega008.03 2007-10-21

Última revisión para la entrega 8, pendiente restricciones de agregar o quitar componentes.

Revisión Entrega009.01 2007-10-24

Documentadas las restricciones para cambios de agregar o quitar y mejoras en la documentación de las restricciones.

Revisión Entrega009.02 2007-11-04

Última revisión entrega.

1. Introducción

1.1. Tipos de entidades a almacenar

La base de datos deberá almacenar una serie de entidades para permitir la persistencia de la información del sistema.

Objetos

Objetos reales del dominio del sistema, en esta etapa podemos identificar *artículos*, *informes* y *operaciones*.

Plantillas

Entidades utilizadas para definir los *objetos* (sus propiedades, características, posibilidades, etc). Podemos identificar *plantillas de artículos* y otras entidades que las modifican: *partes de plantillas de artículos* y *atributos de plantillas de artículos*.

Cambios

Llamamos *cambios* a las transiciones de estado de las propiedades mas básicas de los *objetos*, podemos identificar *cambios de creación de artículos*, *cambios de atributos de artículos*, *cambios de partes de artículos*.

Los *cambios* deberán ser lo suficientemente descriptivos, y estar agrupados de forma tal forma, que permitan obtener eficientemente el estado del sistema en cualquier momento indicado.

1.2. Convenciones de nombres

Dado que la nomenclatura de los elementos de la base de datos en los minisistemas anteriores fue tornándose cada vez mas anárquica¹ se intentará definir una convención mas o menos lógica para los nombres.

Nombres de las tablas

1. Los nombres de las tablas comenzarán con la letra “T”, seguida de otra letra que indique - dentro de lo posible - a que tipo de entidades estará asociado su contenido y luego un carácter de guión bajo “_”.

“TO_”

Para las tablas relacionadas con los objetos del dominio del sistema propiamente dichos, como los artículos.

“TP_”

Para las tablas utilizadas para definir las plantillas.

“TC_”

Para las tablas que mantengan información acerca de los *cambios*.

2. A continuación, un nombre descriptivo (en plural) del tipo de concepto asociado a la tabla, por ejemplo: TO_ARTICULOS, TP_ARTICULOS o TO_INFORMES.

Si la tabla tiene que ver con una relación, se agruparan los nombres de los conceptos asociados, separados nuevamente por el símbolo “_”. Si en este caso los nombres de los conceptos son muy largos, se abreviarán como se crea conveniente, omitiendo todo de intento de incorporar la idea de pluralidad, por ejemplo: TP_ATRIB_ART o TC_PARTE_ART.

Nombres de los campos o parámetros

1. El nombre de los identificadores, ya sean claves primarias o externas, se compondrá de los símbolos “ID_” seguido del concepto asociado, por ejemplo: ID_ARTICULO.
2. Excepción a la regla anterior será para los identificadores de las plantillas, cuyo nombre se compondrá de los símbolos “IDP_” seguidos del concepto asociado, por ejemplo: IDP_ARTICULO.
3. En los casos de que el identificador compuesto de la forma anterior no sea lo suficientemente descriptivo - consideremos el caso de una *parte* que está asociada a un *artículo continente* y un *artículo contenibile* - el nombre de las claves externas podrá tomar otro nombre, por ejemplo: ID_CONTINENTE.
4. Los nombres de los campos que contengan un identificador amigable al usuario de los contenidos de una tabla, por ejemplo el nombre de una plantilla de artículo, comenzarán con los símbolos “NB_”, o “NBP_” si se trata de una plantilla; por ejemplo: NB_PARTE o NBP_ARTICULO.
5. Como en el caso de las claves, si fuera conveniente podrán utilizarse nombres mas específicos, por ejemplo: NBP_CONTINENTE.
6. Los campos que indiquen valores por defecto siempre tomarán el nombre de DEFECTO
7. Los campos que indiquen que el registro fue eliminado tendrán el nombre de ELIMINADO

Nombres de los stored procedures

1. Los nombres de los stored procedures comenzarán con las letras “SP”, seguida de otra letra que indique - dentro de lo posible - a que tipo de entidades estará asociado su contenido y, luego, de un carácter de guión bajo “_”:

“SPO_”

Para los stored procedures relacionados con los objetos del dominio del sistema propiamente dichos, como los artículos.

“SPP_”

Para stored procedures relacionados con las plantillas.

“SPC_”

Para los stored procedures relacionados con los *cambios*.

“SPE_”

Para los stored procedures relacionados con los *estados* derivados de los *cambios*.

2. A continuación una cadena de tres caracteres que indique el tipo de acción que realizará el stored procedure seguida de otro carácter de guión bajo “_”:

“SP?_NEW_”

Para crear elementos.

“SP?_DEL_”

Para eliminar elementos.

“SP?_EDT_”

Para editar elementos.

“SP?_LST_”

Para listar elementos.

“SP?_REN_”

Para renombrar elementos.

“SP?_GET_”

Para recuperar algún valor específico.

“SP?_VRF_”

Para verificar “*algo*”, en caso de fallar la verificación disparará una excepción.

3. A continuación un nombre descriptivo del tipo de concepto asociado a la tabla.

Otros elementos

- Los nombres de las *excepciones* comenzarán con los caracteres “EX_”.
- Los nombres de los *dominios* comenzarán con los caracteres “D_”.
- Los nombres de los *generadores* comenzarán con los caracteres “GEN_”.
- Los nombres de los roles comenzarán con los caracteres “ROL_”.
- Los nombres de las vistas serán similares a los de las tablas pero precedidos por la letra “V”.

2. Descripción de la base de datos

2.1. Tipos de Datos / Dominios

Los campos de las tablas de la base de datos se definirán según dominios predefinidos. El uso de los mismos, además de permitir indicar el tipo de datos, dará una idea del sentido del valor almacenado y brindará otro nivel de abstracción.

D_BOOLEANO

Un valor definido por este dominio representa los conceptos de *SI* o *NO*.

D_DESCRIPCION

Define a las cadenas de caracteres utilizadas para las descripciones de los objetos.

D_ID

Define a los identificadores numéricos de los objetos o conceptos.

D_NOMBRE

Define a los identificadores amigables de los objetos o conceptos.

D_TIPO_OPERACION

Define a los tipos de operaciones (ver *It5-Modelo_Conceptual* [PDF]).

D_TIPO_CMB_PARTE

Define a los tipos de cambios de partes posibles: *AgregarComponente* o *QuitarComponente*.

D_VALOR_ATRIB

Define a los valores de los atributos, es un tipo de datos *VARIABLE*, ya que puede tomar distintos tipos de valores según corresponda al tipo del atributo. ²

D_TIPO_ATRIB

Define a los tipos posibles de D_VALOR_ATRIB.

D_USER_NAME

Define a los nombres e identificadores de los usuarios del sistema.

D_FECHA

Define a las fechas admitidas por el sistema.

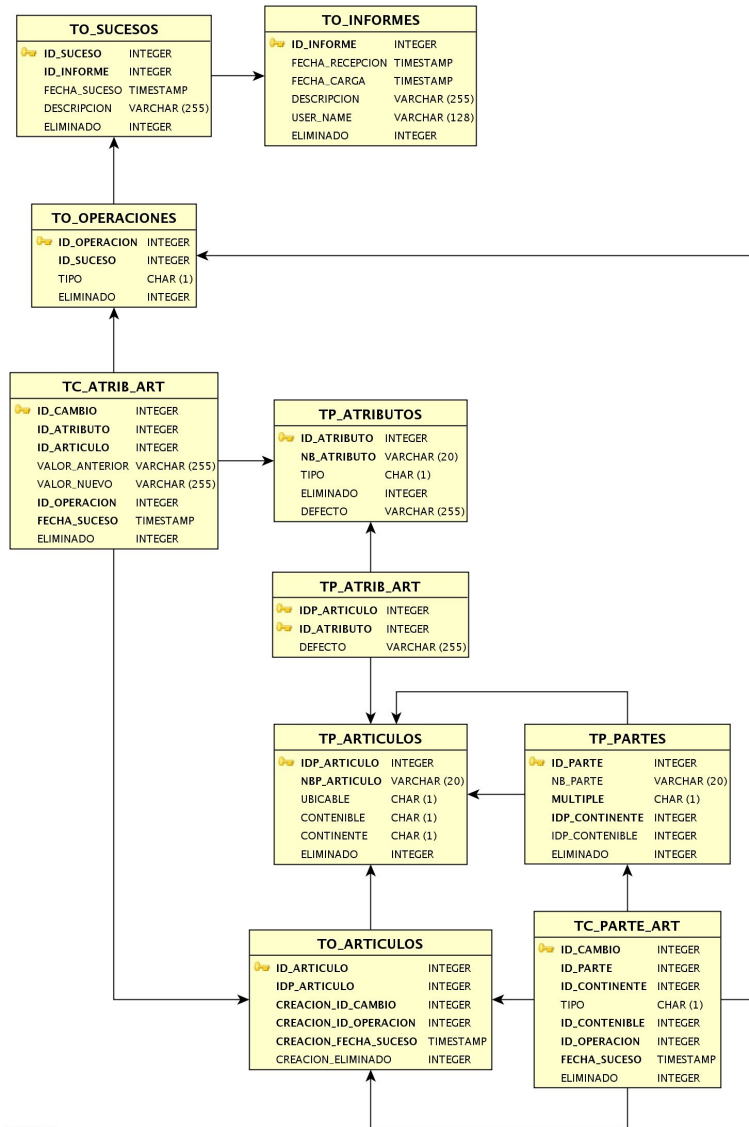
D_ELIMINADO

Define al tipo de datos que indicará que un elemento ya no deberá ser utilizado o listado.

2.2. Entidades / Tablas

El sistema deberá almacenar información relativa a diferentes entidades distribuyendo su información en diferentes tablas.

2.2.1. Diagrama de base de datos



2.2.2. Plantillas de artículos

La información relacionada con las plantillas de artículos, luego de ser normalizada, deberá almacenarse en una serie de tablas que podrían ser las siguientes:

TP_ARTICULOS

Describe las plantillas de artículos.

Campo: IDP_ARTICULO

Dominio: D_ID

Descripción: PK

Campo: NBP_ARTICULO

Dominio: D_NOMBRE

Descripción: Identificador amigable, para hacer referencia a la plantilla desde fuera de la base de datos.

Campo: CONTENIBLE

Dominio: D_BOOLEANO

Descripción: Indica si la plantilla describe artículos de tipo contenible.

Campo: CONTINENTE

Dominio: D_BOOLEANO

Descripción: Indica si la plantilla describe artículos de tipo continente.

Campo: ELIMINADO

Dominio: D_ELIMINADO

Descripción: Indica si la plantilla ya no debe utilizarse.

TP_PARTES

Describe las posibles partes de las plantillas de artículos.

Campo: ID_PARTE

Tipo: D_ID

Descripción: PK

Campo: NB_PARTE

Tipo: D_NOMBRE

Descripción: Identificador amigable, para referenciar la parte desde fuera de la base de datos.

Campo: MULTIPLE

Tipo: D_BOOLEANO

Descripción: Indica si la parte podría contener mas de un artículo contenible simultáneamente o no.

Campo: IDP_CONTINENTE

Tipo: D_ID

Descripción: Indica a que tipo de artículos pertenecería la parte.

Campo: IDP_CONTENIBLE

Tipo: D_ID

Descripción: Indica que tipo de artículos pueden conformar la parte.

Campo: ELIMINADO

Tipo: D_ELIMINADO

Descripción: Indica que la definición de parte ya no debe utilizarse.

TP_ATRIBUTOS

Describe los posibles atributos posibles para cualquier artículo del dominio del sistema.

Campo: ID_ATRIBUTO

Dominio: D_ID

Descripción: PK

Campo: NB_ATRIBUTO

Dominio: D_NOMBRE

Descripción: Identificador amigable, para hacer referencia al atributo desde fuera de la base de datos.

Campo: TIPO

Dominio: D_TIPO_ATRIB

Descripción: Describe el tipo de valor que representa el atributo, por ejemplo: moneda, cantidad, fecha o texto.

Campo: ELIMINADO

Dominio: D_ELIMINADO

Descripción: Indica que la definición del atributo ya no debe utilizarse.

Campo: DEFECTO

Dominio: D_VALOR_ATRIB

Descripción: Indica el valor por defecto que adoptará el atributo al asociarse a una plantilla de artículos.

TP_ATRIB_ART

Representa las asociaciones entre atributos y plantillas de artículos, es decir, los atributos que tendrán los artículos definidos por cada plantilla.

Campo: IDP_ARTICULO

Dominio: D_ID

Descripción: Identificador de la plantilla de artículos.

Campo: ID_ATRIBUTO

Dominio: D_ID

Descripción: Identificador del atributo asociado.

Campo: DEFECTO

Dominio: D_VALOR_ATTRIB

Descripción: Indica el valor por defecto que adoptará el atributo para los artículos nuevos definidos por la plantilla.

2.2.3. Informes

La información de los informes está muy relacionada con la información de transiciones de los estados de las entidades del dominio del sistema, al ser normalizada, podría distribuirse en las siguientes tablas:

TO_INFORMES

Representa los informes ingresados al sistema.

Campo: ID_INFORME

Dominio: D_ID

Descripción: PK

Campo: FECHA_RECEPCION

Dominio: D_FECHA

Descripción: Indica la fecha en que se recibió el informe.

Campo: FECHA_CARGA

Dominio: D_FECHA

Descripción: Indica la fecha se cargó el informe en el sistema.

Campo: DESCRIPCION

Dominio: D_DESCRIPCION

Descripción: Cadena de texto que permitirá relacionar los informes con el informe real, por ejemplo indicando quien lo realizó o a que número de talonario pertenece.

Campo: USER_NAME

Dominio: D_USER_NAME

Descripción: Permitirá identificar al usuario responsable de cargar el informe al sistema.

Campo: ELIMINADO

Dominio: D_ELIMINADO

Descripción: Si un informe fue introducida por error, este podrá marcarse como eliminado, lo cual hará que los sucesos relacionadas con el informe queden sin efecto. Eso solo será posible si se cumplen las reglas de integridad temporal definidas en *It5-Disenio* [PDF]

TO_SUCESOS

Representa un conjunto de operaciones que se realizaron como una unidad.

Campo: ID_SUCESO

Dominio: D_ID

Descripción: PK

Campo: ID_INFORME

Dominio: D_ID

Descripción: Vínculo con el informe que contiene el suceso.

Campo: FECHA_SUCESO

Dominio: D_FECHA

Descripción: Fecha en la que (según se informó) aconteció el suceso.

Campo: DESCRIPCION

Dominio: D_DESCRIPCION

Descripción: Breve descripción del suceso.

Campo: ELIMINADO

Dominio: D_ELIMINADO

Descripción: Si un suceso fue introducida por error, este podrá marcarse como eliminado, lo cual hará que las operaciones relacionadas con el suceso queden sin efecto. Eso solo será posible si se cumplen las reglas de integridad temporal definidas en *It5-Disenio* [PDF]

TO_OPERACIONES

Representa las operaciones descritas en los informes.

Campo: ID_OPERACION

Dominio: D_ID

Descripción: PK

Campo: ID_SUCESO

Dominio: D_ID

Descripción: Vínculo con el suceso que contiene la operación.

Campo: TIPO

Dominio: D_TIPO_OPERACION

Descripción: Indica que el tipo de operación que representa.

Campo: ELIMINADO

Dominio: D_ELIMINADO

Descripción: Si una operación fue introducida por error, esta podrá marcarse como eliminada, lo cual hará que los cambios relacionados queden sin efecto. Eso solo será posible si se cumplen las reglas de integridad temporal definidas en *It5-Disenio* [PDF]

2.2.4. Artículos

Parte de la información de los artículos es heredada de la plantilla que los define, por lo tanto en las tablas relacionadas con la definición de las plantillas hay parte de la información de los artículos, por ejemplo, que atributos tienen.

Otra parte de la información de los artículos es la relacionada con el estado de sus partes y atributos, estos datos podrán ser inferidos de las tablas de cambios.

TO_ARTICULOS

Representa los artículos controlados por el sistema. También el cambio de ingreso del artículo, ya que solo ocurre una vez en la vida del mismo.

Campo: ID_ARTICULO

Dominio: D_ID

Descripción: PK. También será el identificador que utilizarán los usuarios para hacer referencia a los artículos.

Campo: IDP_ARTICULO

Dominio: D_ID

Descripción: Vínculo con la plantilla que define al artículo.

Campo: CREACION_ID_CAMBIO

Dominio: D_ID

Descripción: Identificador global autoincremental de cambio.

Campo: CREACION_ID_OP

Dominio: D_ID

Descripción: Vínculo con la plantilla que define al artículo.

Campo: CREACION_FECHA

Dominio: D_FECHA

Descripción: Correspondiente a la FECHA_SUCESO de creación del artículo.

Campo: CREACION_ELIMINADO

Dominio: D_ELIMINADO

Descripción: Si un artículo fue creado por error, el cambio de creación podrá marcarse como eliminado, lo cual hará que quede sin efecto. Eso solo será posible si se cumplen las reglas de integridad temporal definidas en *It5-Disenio* [PDF]

TC_PARTE_ART

Describe los cambios relacionados con transiciones en las asociaciones entre los artículos continentes y los contenibles que integran sus partes.

Campo: ID_CAMBIO

Dominio: D_ID

Descripción: PK

Campo: ID_OPERACION

Dominio: D_ID

Descripción: Vínculo con la operación que generó el cambio.

Campo: ID_PARTE

Dominio: D_ID

Descripción: Vínculo entre el cambio y la definición de la parte.

Campo: ID_CONTINENTE

Dominio: D_ID

Descripción: Identificador del artículo continente al que se le agregan o quitan componentes.

Campo: TIPO

Dominio: D_TIPO_CMB_PARTE

Descripción: Indica si el cambio representa una adición o una sustracción del componente.

Campo: ID_CONTENIBLE

Dominio: D_ID

Descripción: Identificador del contenible agregado o quitado,

Campo: FECHA_SUCESO

Dominio: D_FECHA

Descripción: Fecha en la que se supone que ocurrió el cambio de parte.

TC_ATTRIB_ART

Describe los cambios relacionados con transiciones en los atributos de los artículos.

Campo: ID_CAMBIO

Dominio: D_ID

Descripción: PK

Campo: ID_OPERACION

Dominio: D_ID

Descripción: Vínculo con la operación que generó el cambio.

Campo: ID_ATRIBUTO

Dominio: D_ID

Descripción: Vínculo entre el cambio y la definición del Atributo.

Campo: ID_ARTICULO

Dominio: D_ID

Descripción: Identificador del artículo al que le vario el valor del atributo.

Campo: VALOR_ANTERIOR

Dominio: D_VALOR_ATTRIB

Descripción: Valor anterior del atributo indicado del artículo indicado.

Campo: VALOR_NUEVO

Dominio: D_VALOR_ATTRIB

Descripción: Nuevo valor del atributo indicado del artículo indicado.

Campo: FECHA_SUCESO

Dominio: D_FECHA

Descripción: Fecha en la que se supone que ocurrió el cambio de valor del atributo.

3. Operaciones, cambios y estados

3.1. Obtener los estados

En *It5-Diseño* [PDF] se explica el procedimiento para obtener los estados a partir de los cambios. Adaptando dicho procedimiento a nuestra estructura de base de datos obtenemos los procedimientos adecuados para cada tipo de estado.

3.1.1. Estado de existencia de artículos

Si bien en este momento el stock de artículos se corresponde directamente con la tabla TO_ARTICULOS ya que los artículos solo pueden ingresar al sistema una única vez y, aún, no es posible darlos de baja. Ya podría esbozarse un stored procedure para obtener los artículos activos en un momento dado.

```

/* Obtener el listado de artículos activos en un
 * instante determinado */
create procedure SPE_LST_ID_ARTICULOS (
    :FECHA D_FECHA)
returns (
    :ID_OPERACION D_ID,
    :FECHA_SUCESO D_FECHA,
    :ID_ARTICULO D_ID)
as
begin
    for select
        TO_ARTICULOS.CREACION_ID_OP,
        TO_ARTICULOS.CREACION_FECHA,
        TO_ARTICULOS.ID_ARTICULO
    from
        TO_ARTICULOS
    where
        TO_ARTICULOS.CREACION_ELIMINADO = NO and
        ((:FECHA is null) or (TO_ARTICULOS.CREACION_FECHA <= :FECHA))
    into
        :ID_OPERACION,
        :FECHA_SUCESO,
        :ID_ARTICULO
    do
        suspend;
    exit;
end;

```

3.1.2. Estados de los atributos de los artículos

Los estados de los atributos de los artículos existentes corresponden al valor asignado a dichos atributos. En cada momento cada atributo de cada artículo tiene un y solo un valor.

```

/* Obtener el valor de un atributo de un artículo en un
 * instante determinado. */
create procedure SPE_GET_VALOR_ATRIB_ART (
    :ID_ARTICULO D_ID,
    :ID_ATRIBUTO D_ID,
    :FECHA D_FECHA)
returns (
    :ID_OPERACION D_ID,
    :FECHA_SUCESO D_FECHA,
    :VALOR D_VALOR_ATRIB)
as
begin
    select first 1
        TC_ATRIB_ART.VALOR_NUEVO,

```

```

        TC_ATRIB_ART.FECHA_SUCEO,
        TC_ATRIB_ART.ID_OPERACION
    from
        TC_ATRIB_ART
    where
        TC_ATRIB_ART.ELIMINADO = NO and
        TC_ATRIB_ART.FECHA_SUCEO <= :FECHA and
        TC_ATRIB_ART.ID_ARTICULO = :ID_ARTICULO and
        TC_ATRIB_ART.ID_ATRIBUTO = :ID_ATRIBUTO
    order by
        TC_ATRIB_ART.ID_CAMBIO descending
    into
        :VALOR,
        :FECHA_SUCEO,
        :ID_OPERACION
    exit;
end;

/* Listar los valores de los atributos de los
 * artículos en un instante determinado. */
create procedure SPE_LST_VALORES_ATRIB_ART (
    :FECHA D_FECHA)
returns (
    :ID_OPERACION D_ID,
    :FECHA_SUCEO D_FECHA,
    :ID_ARTICULO D_ID,
    :ID_ATRIBUTO D_ID,
    :VALOR D_VALOR_ATRIB)
as
begin
    for select
        TO_ARTICULOS.ID_ARTICULO,
        TP_ATRIB_ART.ID_ATRIBUTO
    from
        SPE_LST_ID_ARTICULOS(:FECHA)
    inner join TO_ARTICULOS on TO_ARTICULOS.ID_ARTICULO = SPE_LST_ID_ARTICULOS.ID_ARTICULO
    inner join TP_ATRIB_ART on TP_ATRIB_ART.IDP_ARTICULO = TO_ARTICULOS.IDP_ARTICULO
    into
        :ID_ARTICULO,
        :ID_ATRIBUTO
    do
    begin
        execute procedure SPE_GET_VALOR_ATRIB_ART (
            :FECHA,
            :ID_ARTICULO,
            :ID_ATRIBUTO)
        returningvalues (
            :ID_OPERACION,
            :FECHA_SUCEO,
            :VALOR);
        suspend;
    end
    exit;
end;
end;

```

3.1.3. Estados de los artículos contenibles (contenidos / no contenidos)

Los artículos contenibles pueden estar contenidos (asociados a una parte de un continente) o no contenidos.

Si un continente nunca se asoció a una parte, naturalmente, estará no contenido. Si ya se ha asociado a alguna parte, su estado de contenido o no contenido dependerá de si el último movimiento fue una asociación o una des-asociación.

```

/* Obtener el último movimiento (asociación o desasociación)
 * de un artículo contenible un instante determinado. */
create procedure SPE_GET_PARTE_CONTE (
    :ID_CONTENIBLE D_ID,
    :FECHA D_FECHA)
returns (
    :TIPO D_TIPO_CMB_PARTE,
    :ID_PARTE D_ID,
    :ID_CONTINENTE D_ID,
    :FECHA_SUCESO D_FECHA,
    :ID_OPERACION D_ID)
as
begin
    select
        first(TC_PARTE_ART.TIPO),
        first(TC_PARTE_ART.ID_PARTE),
        first(TC_PARTE_ART.ID_CONTINENTE),
        first(TC_PARTE_ART.FECHA_SUCESO),
        first(TC_PARTE_ART.ID_OPERACION)
    from
        TC_PARTE_ART
    where
        TC_PARTE_ART.ELIMINADO = NO and
        TC_PARTE_ART.FECHA_SUCESO <= :FECHA and
        TC_PARTE_ART.ID_CONTENIBLE = :ID_CONTENIBLE
    order by
        TC_PARTE_ART.ID_CAMBIO descending
    group by
        TC_PARTE_ART.ID_CONTENIBLE
    into
        :TIPO,
        :ID_PARTE,
        :ID_CONTINENTE,
        :FECHA_SUCESO,
        :ID_OPERACION;
    exit;
end;

/* Listar los artículos contenibles no contenidos
 * en un instante determinado. */
create procedure SPE_LST_CONTE_NOCONTENIDOS (
    :FECHA D_FECHA)
returns (
    :ID_CONTENIBLE D_ID,
    :FECHA_SUCESO D_FECHA,
    :ID_OPERACION D_ID)
as
begin
    for select
        TO_ARTICULOS.ID_ARTICULO,
        TO_ARTICULOS.CREACION_FECHA,
        TO_ARTICULOS.CREACION_ID_OP
    from
        TO_ARTICULOS
        inner join TP_ARTICULOS
            on TO_ARTICULOS.IDP_ARTICULO = TP_ARTICULOS.IDP_ARTICULO
    where
        TP_ARTICULOS.CONTENIBLE = SI
    Into
        :ID_CONTENIBLE,
        :CREACION_FECHA,
        :CREACION_ID_OP
    do
        begin

```

```

execute procedure SPE_GET_PARTE_CONTE (
    :ID_CONTENIBLE,
    :FECHA)
returning_values (
    :TIPO,
    :FECHA_SUCESO,
    :ID_OPERACION);
if (:TIPO is null) then
begin
    FECHA_SUCESO = :CREACION_FECHA
    ID_OPERACION = :CREACION_ID_OP
    suspend;
end
else if (:TIPO = QuitarComponente)) Then
    suspend;
end
exit;
end;

/* Listar los artículos contenibles contenidos
 * en un instante determinado. */
create procedure SPE_LST_CONTE_CONTENIDOS (
    :FECHA D_FECHA)
returns (
    :ID_CONTENIBLE D_ID,
    :ID_PARTE D_ID,
    :ID_CONTINENTE D_ID
    :FECHA_SUCESO D_FECHA,
    :ID_OPERACION D_ID)
as
begin
    for select
        TO_ARTICULOS.ID_ARTICULO
    from
        TO_ARTICULOS
    inner join TP_ARTICULOS
        on TO_ARTICULOS.IDP_ARTICULO = TP_ARTICULOS.IDP_ARTICULO
    where
        TP_ARTICULOS.CONTENIBLE = SI
    Into
        :ID_CONTENIBLE
    do
begin
    execute procedure SPE_GET_PARTE_CONTE (
        :ID_CONTENIBLE,
        :FECHA)
    returning_values (
        :TIPO,
        :ID_PARTE,
        :ID_CONTINENTE,
        :FECHA_SUCESO,
        :ID_OPERACION);
    if ((:TIPO is not null) and (:TIPO = AgregarComponente)) Then
        suspend;
end
exit;
end;

```

3.1.4. Estados de las partes de artículos continentes

El estado de las partes de los artículos continentes dependerá de si la parte es simple o múltiple.

Si es simple, corresponde al contenible asociado a dicha parte y estará definido por el último movimiento asociado a dicha parte.

Mientras que si es múltiple, corresponde a la lista de artículos asociados a dicha parte y estará definido por los últimos movimientos de cada contenible asociado.

```

/* Obtener el estado de una
 * parte simple de un artículo continente. */
create procedure SPE_GET_PARTE_CONTI_SIMPLE (
    :ID_CONTINENTE D_ID,
    :ID_PARTE D_ID,
    :FECHA D_FECHA)
returns (
    :ID_CONTENIBLE D_ID,
    :FECHA_SUCESO D_FECHA,
    :ID_OPERACION D_ID)
as
begin
    select
        case first(TC_PARTE_ART.TIPO)
            when AgregarComponente then
                first(TC_PARTE_ART.ID_CONTENIBLE)
            when QuitarComponente then
                null
        end,
        first(TC_PARTE_ART.FECHA_SUCESO),
        first(TC_PARTE_ART.ID_OPERACION)
    from
        TC_PARTE_ART
    where
        TC_PARTE_ART.ELIMINADO = NO and
        TC_PARTE_ART.FECHA_SUCESO <= :FECHA and
        TC_PARTE_ART.ID_CONTINENTE = :ID_CONTINENTE and
        TC_PARTE_ART.ID_PARTE = :ID_PARTE
    order by
        TC_PARTE_ART.ID_CAMBIO descending
    group by
        TC_PARTE_ART.ID_CONTINENTE,
        TC_PARTE_ART.ID_PARTE
    into
        :ID_CONTENIBLE,
        :FECHA_SUCESO,
        :ID_OPERACION;

    if (:FECHA_SUCESO is null) then
    begin
        select
            TO ARTICULOS.CREACION_FECHA,
            TO ARTICULOS.CREACION_ID_OP
        from
            TO ARTICULOS
        where
            TO ARTICULOS.ID_ARTICULO = ID_CONTINENTE
        into
            :FECHA_SUCESO,
            :ID_OPERACION;
    end;
    exit;
end;

```

/* Obtener el estado de una

```

* parte multiple de un artículo continente. */
create procedure SPE_GET_PARTE_CONTI_MULTIPLE (
    :ID_CONTINENTE D_ID,
    :ID_PARTE D_ID
    :FECHA D_FECHA)
returns (
    :ID_CONTENIBLE D_ID,
    :FECHA_SUCESO D_FECHA,
    :ID_OPERACION D_ID)
as
begin
    for select
        :ID_CONTENIBLE D_ID,
        first(TC_PARTE_ART.TIPO),
        first(TC_PARTE_ART.FECHA_SUCESO),
        first(TC_PARTE_ART.ID_OPERACION)
    from
        TC_PARTE_ART
    where
        TC_PARTE_ART.ELIMINADO = NO and
        TC_PARTE_ART.FECHA_SUCESO <= :FECHA and
        TC_PARTE_ART.ID_CONTINENTE = :ID_CONTINENTE and
        TC_PARTE_ART.ID_PARTE = :ID_PARTE
    order by
        TC_PARTE_ART.ID_CAMBIO descending
    group by
        TC_PARTE_ART.ID_CONTINENTE,
        TC_PARTE_ART.ID_CONTENIBLE,
        TC_PARTE_ART.ID_PARTE
    into
        :ID_CONTENIBLE,
        :TIPO,
        :FECHA_SUCESO,
        :ID_OPERACION
    do
        if (:TIPO = AgregarComponente) then
            suspend;
        exit;
    end;
end;

```

3.2. Restricciones

Mas allá de la integridad referencial, deberán implementarse otros controles para garantizar la validez e integridad de los datos almacenados en la base de datos.

Ya que el modelo relacional actualmente no maneja la temporalidad, muchas de las restricciones se relacionarán con el control de la coherencia de los datos temporales y se implementarán mediante triggers.

Deberán revisarse la coherencia tanto en el momento de insertar datos, como en el de la eliminación o anulación.

Plantillas de artículos

El objeto de las plantillas se limita a facilitar la creación y descripción de los “verdaderos” elementos del sistema, como son los artículos. Por lo tanto, se intentará que no generen demasiadas restricciones; por ejemplo, si el día de hoy se agrega un atributo a determinada plantilla de artículos, no debería haber inconveniente en modificar el valor del atributo de un artículo perteneciente a dicha plantilla con fecha anterior a la actual.

- Al asignar un atributo a una plantilla de artículos, generar los cambios de inicialización del atributo para cada uno de los artículos de la plantilla asociados al suceso de creación de dicho artículo (como si el atributo hubiera existido desde siempre).

Implementada en:

- TP_ATRIB_ART_AI0

Creación de artículos

- Al crear artículos se deberá generar un cambio de inicialización para cada uno de sus atributos, con el objeto de setear el valor por defecto indicado en su plantilla.

Implementada en:

- Trigger TO_ARTICULOS_AI0.

- La fecha del cambio de creación de un artículo siempre deberá corresponder a la fecha del suceso que la generó.

Implementada en:

- Trigger TO_ARTICULOS_BI.

Cambio de valor de un atributo de un artículo

- La fecha del cambio de un atributo de un artículo siempre deberá corresponder a la fecha del suceso que la generó.

Implementada en:

- Trigger TC_ATRIB_ART_BI.

- En la fecha del suceso de cambio del atributo debe existir el artículo.

Implementada en:

- Trigger TC_ATRIB_ART_BI.

- La plantilla del artículo debe tener el atributo indicado.

Implementada en:

- Trigger TC_ATRIB_ART_BI.

- Debe ser valido con respecto a otros cambios ingresados:

Si el cambio corresponde a la creación de un artículo, el valor anterior debe ser indefinido.

Si el cambio no corresponde a la creación de un artículo, el valor anterior indicado puede ser indefinido (en ese caso adoptará el adecuado), o bien, debe corresponderse con el ultimo valor de ese atributo para ese artículo.

Implementada en:

- Trigger TC_ATRIB_ART_BI.

Agregar o quitar componentes

- La fecha del cambio de agregado o quitado de componentes siempre deberá corresponder a la fecha del suceso que la generó.

Implementada en:

- Trigger TC_PARTE_ART_BI.

- En la fecha del suceso deben existir los dos artículos.

Implementada en:

- Trigger TC_PARTE_ART_BI.

- La plantilla del artículo continente debe tener la parte indicada y debe corresponder con la plantilla del artículo contenible.

Implementada en:

- Trigger TC_PARTE_ART_BI.

- Debe ser valido con respecto a otros cambios ingresados:

Si se está agregando un componente, debe verificarse que el contenible y la parte correspondiente del continente estén disponibles.

Si se está quitando un componente, debe verificarse que el contenible esté asociado a la parte indicada del continente.

Implementada en:

- Trigger TC_PARTE_ART_BI.

Notas

1. A punto tal de que los nombres dejaron de describir, siquiera remotamente, la funcionalidad.
2. Una posible implementación para este dominio sería utilizar campos varchar y convertir el valor según sea el D_TIPO_ATRIB asociado. Otra posible implementación sería utilizar mas de una tabla para almacenar los valores asociados a este dominio: una tabla para cada posible tipo.