



## **MANUAL PARA ESTÁNDARES DE MODELADO DE BASE DE DATOS RELACIONAL**

Aprobado con Resolución Administrativa  
AGETIC/RA/0101/2022, de 12 de agosto de 2022

**UNIDAD DE PRODUCCIÓN Y ACTUALIZACIÓN  
TECNOLÓGICA**

**(UPAT)**



## MANUAL METODOLÓGICO PARA LA GESTIÓN DE RIESGOS Y OPORTUNIDADES EN LA AGETIC

Código: UPAT/ADET-M03

Versión: 0

Aprobado: R.A. AGETIC/RA/0101/2022, de 12/08/2022

## CONTENIDO

CAPÍTULO I.....	3
ASPECTOS GENERALES.....	3
1. Objeto.....	3
2. Marco Normativo.....	3
3. Alcance y/o Ámbito de Aplicación.....	3
4. Previsión.....	4
5. Definiciones.....	4
6. Aprobación, Vigencia, Difusión e Implementación del Manual.....	4
7. Revisión y Actualización.....	5
CAPÍTULO II.....	6
ESTÁNDARES DE MODELADO DE BASE DE DATOS.....	6
8. Estándares de Base de Datos.....	6
9. Convenciones Generales.....	6
10. Objetos de Base de Datos.....	8

	<b>MANUAL METODOLÓGICO PARA LA GESTIÓN DE RIESGOS Y OPORTUNIDADES EN LA AGETIC</b>	
Código: UPAT/ADET-M03	Versión: 0	Aprobado: R.A. AGETIC/RA/0101/2022, de 12/08/2022

## CAPÍTULO I ASPECTOS GENERALES

### 1. Objeto

Definir los estándares aplicables al proceso de diseño del modelo de datos y su representación en bases de datos relacionales, como parte de los productos de software dentro del Área de Desarrollo Tecnológico (ADET) dependiente de la Unidad de Producción y Actualización Tecnológica (UPAT) de la AGETIC.

### 2. Marco Normativo

Los estándares, tiene como marco normativo las siguientes disposiciones:

- a. Constitución Política del Estado, 7 de febrero de 2009.
- b. Ley N° 1178, de 20 de julio de 1990, de Administración y Control Gubernamentales.
- c. Decreto Supremo N° 23318-A, de 3 de noviembre de 1992, Reglamento de la Responsabilidad por la Función Pública.
- d. Decreto Supremo N° 25350, de 8 de abril de 1999, Manual de Técnicas Normativas.
- e. Decreto Supremo N° 26237, de 29 de junio del 2001, Reglamento de Responsabilidad por la Función Pública y sus modificaciones.
- f. Decreto Supremo N° 2514, de 9 de septiembre de 2015, crea la Agencia de Gobierno Electrónico y Tecnologías de Información y Comunicación - AGETIC.
- g. Resolución Suprema N° 217055, de 20 de mayo de 1997, Normas Básicas del Sistema de Organización Administrativa.
- h. Decreto Supremo N° 1793, de 13 de noviembre de 2013, Reglamento para el Desarrollo de Tecnologías de Información y Comunicación.
- i. Resolución Administrativa AGETIC/RA/0084/2022, de 19 de julio de 2022, que aprueba el documento "MANUAL DE PROCEDIMIENTOS DE DESARROLLO Y ACTUALIZACIÓN DE PRODUCTOS DE SOFTWARE DE LA AGETIC".
- j. Resolución Administrativa AGETIC/RA/0042/2022, de 26 de mayo de 2022, que aprueba el documento "MANUAL DE POLÍTICAS, ESTÁNDARES Y LINEAMIENTOS DE DESARROLLO Y ACTUALIZACIÓN DE SOFTWARE".

### 3. Alcance y/o Ámbito de Aplicación

El presente Manual es de aplicación para todo el personal con ítem y consultores de línea de la Unidad de Producción y Actualización Tecnológica (UPAT) cuyas actividades están relacionadas con los procesos de desarrollo y actualización de software del Área de Desarrollo Tecnológico (ADET).

	<b>MANUAL METODOLÓGICO PARA LA GESTIÓN DE RIESGOS Y OPORTUNIDADES EN LA AGETIC</b>	
Código: UPAT/ADET-M03	Versión: 0	Aprobado: R.A. AGETIC/RA/0101/2022, de 12/08/2022

#### 4. Previsión

En caso de presentarse dudas, omisiones, contradicciones y/o diferencias en la interpretación de las Políticas, Lineamientos y Estándares del presente Manual, éstas serán solucionadas en los alcances y previsiones establecidas en las disposiciones legales y normativas pertinentes.

#### 5. Definiciones

- a. **Producto de Software**, es una colección integrada de componentes lógicos que se originan a partir de la ejecución de un conjunto de fases -denominado ciclo de vida- a una lista de requerimientos. La característica principal de un producto de software es que puede ser compartido y empaquetado para su distribución y explotación.
- b. **Desarrollo de Software**, se entiende al proceso por el cual se aplica un conjunto de etapas -denominado **ciclo de vida**- a una lista de requerimientos para analizar, diseñar, programar, probar y producir un producto de software.
- c. **Actualización de Software**, es el proceso que se aplica a los productos de software desarrollados, una vez realizada la implementación y despliegue en un ambiente de producción. Normalmente, luego de analizar y dimensionar los requerimientos se procede con el desarrollo, siguiendo el ciclo de vida definido a partir de la etapa de análisis, diseño o programación. La dinámica de actualización no solo se aplica al producto de software funcional sino se aplica también a los entregables, artefactos y componentes creados o modificados por este proceso.
- d. **Estándares**, establecen un conjunto de instrucciones que se deben seguir para lograr un desarrollo y actualización uniforme de los componentes y artefactos de un producto de software.
- e. **Base de Datos Relacional**, es un tipo de base de datos que se encarga de guardar información, teniendo la particularidad que los datos almacenados deben tener una relación a través de sus diversas tablas, filas o columnas.

#### 6. Aprobación, Vigencia, Difusión e Implementación del Manual

El presente manual deberá ser aprobado por el Director General Ejecutivo de la AGETIC mediante Resolución Administrativa.

La vigencia del manual será a partir de la fecha establecida en la Resolución Administrativa de aprobación.

La difusión del manual será realizada por el Área de Planificación (AP) en coordinación con la Unidad de Producción y Actualización Tecnológica (UPAT), siendo de conocimiento general por el personal de la AGETIC.

	<b>MANUAL METODOLÓGICO PARA LA GESTIÓN DE RIESGOS Y OPORTUNIDADES EN LA AGETIC</b>	
Código: UPAT/ADET-M03	Versión: 0	Aprobado: R.A. AGETIC/RA/0101/2022, de 12/08/2022

La implementación del manual será efectuada por la Unidad de Producción y Actualización Tecnológica (UPAT).

## 7. Revisión y Actualización

El presente manual deberá ser ajustado y/o actualizado cuando se produzcan cambios o ajustes en el marco normativo, o cuando por razones internas y/o del entorno se justifique realizar modificaciones.

El Área de Desarrollo Tecnológico (ADET) dependiente de la Unidad de Producción y Actualización Tecnológica (UPAT) es responsable de la actualización del presente manual y se realizará en base a:

- El análisis de la experiencia derivada de su aplicación.
- Las necesidades institucionales que se presenten.
- Los cambios que se hacen en las disposiciones legales.
- Los cambios generados por el Consejo para las Tecnologías de Información y Comunicación del Estado Plurinacional de Bolivia (CTIC-EPB) sobre el documento *“Lineamientos y Estándares de Desarrollo de Software para el rol Desarrollador”*.

El Área de Desarrollo Tecnológico (ADET) en coordinación con el AP realizará el ajuste y actualización del manual cuando se produzcan los cambios señalados.

Toda vez que el manual sea actualizado, deberá darse cumplimiento al numeral precedente de Aprobación, Vigencia, Difusión e Implementación.

	<b>MANUAL METODOLÓGICO PARA LA GESTIÓN DE RIESGOS Y OPORTUNIDADES EN LA AGETIC</b>	
Código: UPAT/ADET-M03	Versión: 0	Aprobado: R.A. AGETIC/RA/0101/2022, de 12/08/2022

## CAPÍTULO II ESTÁNDARES DE MODELADO DE BASE DE DATOS

### 8. Estándares de Base de Datos

Uno de los ámbitos en el desarrollo de sistemas es el diseño del modelo de datos implementado con objetos de base de datos. En este marco es necesario establecer estándares para definir la forma en que se denominan las bases de datos, esquemas, tablas, funciones, tipos y otros objetos asociados a un cluster de base de datos relacional.

### 9. Convenciones Generales

#### Regla EBD01: Denominación de Identificadores

Para la denominación de identificadores de objetos de bases de datos se deben considerar los siguientes puntos:

- Los nombres deben estar en idioma español, sin embargo, se podrán aceptar excepciones cuando se trate de términos en inglés ampliamente aceptados y reconocidos en un contexto tecnológico.
- Todos los nombres deben empezar con letras.
- No se deben utilizar caracteres con tildes ni la letra "ñ".
- Solo se pueden utilizar letras, números y guión bajo (\_).
- Los nombres deben ser los más cortos posibles, pero siempre manteniendo su significado y representatividad.
- No se deben utilizar preposiciones en los nombres, por ejemplo: de, la, el, etc.
- Evitar nombres sin significado o redundantes.

#### Regla EBD02: Uso de Términos en Inglés

Se podrán utilizar términos en inglés ampliamente aceptados en el contexto de las tecnologías utilizadas en el desarrollo de sistemas informáticos. A continuación, se listan varios términos en inglés que podrán mantenerse en ese idioma: *tablespace, using index, defined-user, array, type, trigger, explain*, etc.

#### Regla EBD03: No uso de palabras reservadas

En la nominación de objetos de base de datos, no se pueden utilizar palabras reservadas definidas en gestores de base de datos. Las palabras reservadas interfieren con los

	<b>MANUAL METODOLÓGICO PARA LA GESTIÓN DE RIESGOS Y OPORTUNIDADES EN LA AGETIC</b>	
Código: UPAT/ADET-M03	Versión: 0	Aprobado: R.A. AGETIC/RA/0101/2022, de 12/08/2022

programas desarrollados en PL/pgSQL y otros. Algunos ejemplos son: *select, count, sum, date, numeric, integer, varchar*, etc.

#### **Regla EBD04: Nominación de sistemas**

Los sistemas están asociados a uno o varios esquemas de la base de datos. Todo sistema debe ser identificado por un acrónimo.

El formato es el siguiente:

Variable	Valor
<AcrónimoSistema>	Cadena representativa de hasta 8 caracteres (letras y dígitos) asignado al sistema. El primer carácter siempre debe ser una letra.

Ejemplo:

Acrónimo Sistema	Descripción
seprec	Sistema de Registro de Comercio dependiente del Servicio Plurinacional de Registro de Comercio (SEPREC).
olimat	Sistema de Olimpiadas Matemáticas del Ministerio de Educación de Bolivia.
sidcup	Sistema de Certificaciones Pro-Bolivia

#### **Regla EBD05: Nominación de esquemas**

Los esquemas de la base de datos están asociados a módulos de negocio dentro del sistema. La nominación o nombre del esquema está conformado por un nombre descriptivo corto del esquema.

El formato es el siguiente:

*<NombreEsquema> = <DescripciónEsquema>*

 <b>AGETIC</b> <small>Digitalizando Bolivia</small>	<b>MANUAL METODOLÓGICO PARA LA GESTIÓN DE RIESGOS Y OPORTUNIDADES EN LA AGETIC</b>	
Código: UPAT/ADET-M03	Versión: 0	Aprobado: R.A. AGETIC/RA/0101/2022, de 12/08/2022

Donde:

Variable	Valor
<DescripciónEsquema>	Cadena de hasta 20 caracteres compuesto de letras, dígitos y guiones bajos.

Ejemplos:

Sistema	Nombre Esquema	Descripción
seprec	tramites	Esquema de Trámites
seprec	empresas	Esquema de Empresas
seprec	usuarios	Esquema de Usuarios
seprec	parametricas	Esquema de Paramétricas

Opcionalmente, el acrónimo del sistema podrá formar parte del nombre del esquema, esto estará sujeto a criterio del equipo de desarrollo.

## 10. Objetos de Base de Datos

A continuación se describen los términos utilizados para definir la nomenclatura de los objetos de base de datos.

### 10.1 Tablespaces

#### Regla EBD06: Tablespaces

Los *tablespaces* estarán en directa relación con los esquemas, siendo que por cada esquema se crearán dos (2) tablespaces, uno para datos y otro para índices.

Los nombres de *tablespaces* en el cluster de base de datos estarán definidos por la información que contienen, ya sean tablas de datos o índices de búsqueda.

	<b>MANUAL METODOLÓGICO PARA LA GESTIÓN DE RIESGOS Y OPORTUNIDADES EN LA AGETIC</b>	
Código: UPAT/ADET-M03	Versión: 0	Aprobado: R.A. AGETIC/RA/0101/2022, de 12/08/2022

Los *tablespaces* cuya función sea almacenar tablas de datos tendrán el formato siguiente:

*dat\_<NombreEsquema>*

Los *tablespaces* cuya función sea almacenar índices de búsqueda tendrán el formato siguiente:

*idx\_<NombreEsquema>*

Se crearán carpetas en unidades de disco del sistema operativo (SO) los cuales tendrán una relación directa con los *tablespaces*.

Las carpetas de unidades de disco del SO asociados a *tablespaces* de datos tendrán el formato siguiente:

*data\_<NombreEsquema>*

Las carpetas de unidades de disco del SO asociados a *tablespaces* de índices tendrán el formato siguiente:

*indx\_<NombreEsquema>*

Ejemplos:

```
mkdir /var/disco1/data_tramites
mkdir /var/disco2/indx_tramites
CREATE TABLESPACE dat_tramites
LOCATION '/var/disco1/data_tramites';
CREATE TABLESPACE idx_tramites
LOCATION '/var/disco2/indx_tramites';
```

## 10.2 Base de Datos

### Regla EBD07: Base de Datos

El nombre de la base de datos está relacionado con el nombre del acrónimo de uno o varios sistemas. En la creación se debe especificar un *tablespace* de datos de alguno de sus esquemas.

El formato es el siguiente:



## MANUAL METODOLÓGICO PARA LA GESTIÓN DE RIESGOS Y OPORTUNIDADES EN LA AGETIC

Código: UPAT/ADET-M03

Versión: 0

Aprobado: R.A. AGETIC/RA/0101/2022, de 12/08/2022

<NombreBaseDatos> = <AcrónimoSistema>

Ejemplos:

```
CREATE DATABASE sidcup TABLESPACE dat_usuarios;  
CREATE DATABASE seprec TABLESPACE dat_tramites;
```

### 10.3 Tablas

#### Regla EBD08: Nominación de tablas

Consideraciones básicas de denominación de tablas:

- El nombre de tabla no debe exceder los 60 caracteres.
- La descripción de la tabla siempre estará en plural.
- Se utilizará el carácter guión bajo (\_) entre palabras.

En el caso de tablas que resultan de la relación de otras, el nombre estará compuesto por los nombres (o abreviaciones) de las tablas conformantes.

El formato de nombre de tabla es el siguiente:

<NombreTabla> = <DescripciónTabla>

Ejemplos:

Nombre Esquema	Nombre Tabla
tramites	tramites
tramites	bitacoras
tramites	bitacoras_detalle
empresas	empresas
empresas	empresas_sucursales

 <b>AGETIC</b> <small>Digitalizando Bolivia</small>	<b>MANUAL METODOLÓGICO PARA LA GESTIÓN DE RIESGOS Y OPORTUNIDADES EN LA AGETIC</b>	
Código: UPAT/ADET-M03	Versión: 0	Aprobado: R.A. AGETIC/RA/0101/2022, de 12/08/2022

Toda tabla de datos debe ser creada sobre un *tablespace* destinado para ello, al igual que, todo índice de búsqueda debe ser creado sobre un *tablespace* destinado para ello.

Ejemplo:

```
CREATE TABLE tramites.pagos (
  id          serial8 not null,
  cantidad   integer,
  detalle    varchar(50),
  id_estado  varchar(2),
  PRIMARY KEY (id)
  USING INDEX TABLESPACE idx_tramites
)
tablespace dat_tramites;
```

### **Regla EBD09: Nominación de tablas históricas**

Toda tabla histórica debe llevar el mismo nombre que la tabla base de producción (tabla de datos). Para la agrupación de tablas históricas, previamente, se debe asignar un esquema histórico por cada esquema de producción.

El formato es el siguiente:

*<NombreTablaHistórica> = <NombreTablaBase>*

Ejemplo:

```
Esquema de Producción: tramites
Tabla Base:           detalles
Esquema Histórico:   historico_tramites
Tabla Histórica:     detalles
```

```
CREATE SCHEMA historico_tramites;
CREATE TABLE historico_tramites.detalles AS ...
```

### **Regla EBD10: Nominación de tablas temporales**

Las tablas temporales, tradicionalmente definidas en los ámbitos de sesión o transacción, no están permitidas para su uso.

	<b>MANUAL METODOLÓGICO PARA LA GESTIÓN DE RIESGOS Y OPORTUNIDADES EN LA AGETIC</b>	
Código: UPAT/ADET-M03	Versión: 0	Aprobado: R.A. AGETIC/RA/0101/2022, de 12/08/2022

La funcionalidad de tablas temporales será sustituida por la cláusula WITH de la sentencia SELECT.

Ejemplo:

```

WITH ventas_regionales AS (
  SELECT region, SUM(monto) AS ventas_totales
  FROM ordenes
  GROUP BY region
), regiones_top AS (
  SELECT region
  FROM ventas_regionales
  WHERE ventas_totales >
    (SELECT SUM(ventas_totales)/10 FROM ventas_regionales)
)
SELECT region,
  producto,
  SUM(cantidad) AS unidades_producto,
  SUM(monto) AS ventas_producto
FROM ordenes
WHERE region IN (SELECT region FROM regiones_top)
GROUP BY region, producto;

```

### **Regla EBD11: Nominación de tablas auxiliares**

Las tablas auxiliares almacenan información transitoria.

No deben superar los 60 caracteres de longitud y deben incluir al final del nombre el sufijo AUX, bajo la siguiente forma:

*<NombreTablaAuxiliar> = <DescripciónTabla>\_aux*

Ejemplo:

*CREATE TABLE publicaciones\_aux AS ...*

 <p>AGETIC Digitalizando Bolivia</p>	<b>MANUAL METODOLÓGICO PARA LA GESTIÓN DE RIESGOS Y OPORTUNIDADES EN LA AGETIC</b>	
Código: UPAT/ADET-M03	Versión: 0	Aprobado: R.A. AGETIC/RA/0101/2022, de 12/08/2022

Las tablas auxiliares no mantienen información persistente, por tanto, están sujetas a depuración periódica, y según rutinas de limpieza de base de datos son eliminadas periódicamente.

### Regla EBD12: Comentarios sobre tablas

Toda tabla y sus campos deben incluir comentarios. Esos comentarios deberán reflejar de forma clara y en pocas palabras la utilidad de la tabla y los campos dentro de la misma. Estos comentarios podrán ser registrados al momento de la creación de la tabla, o en su defecto, posterior a ello.

## 10.4 Columnas

### Regla EBD13: Nominación de columnas

Consideraciones básicas de denominación de columnas:

- Los nombres deben ser únicos dentro una tabla.
- Los nombres no deben ser en longitud mayor a 60 caracteres.
- Deben estar escritos en español, o mínimamente tener su significado.
- Deben utilizarse abreviaciones estándar, si corresponde.
- Deben utilizarse correctamente los tipos de datos.
- No se utilizarán en ningún caso los tipos de datos: *char*, *money*, *enum* y *oid*.
- El tipo de dato *timestamp* no debe incluir zona horaria (*without time zone*).
- El uso de los tipos de datos *array* y *user-defined* deben justificarse, y no deben formar parte de columnas de búsqueda en consultas SQL.

Las características no deseables de los tipos de datos restringidos para su uso son los siguientes:

Tipo de Dato	Penalizaciones en su uso
char ( <i>n</i> )	Se rellenará con espacios en blanco las <i>n</i> posiciones de la derecha no utilizadas. En intercalaciones donde los espacios en blanco son importantes, este comportamiento puede producir resultados inesperados.
money	Se desaconseja usar <i>money</i> porque es demasiado sensible a la configuración regional. En su lugar lo que usa la mayoría de los programas

 <b>AGETIC</b> <small>Digitalizando Bolivia</small>	<b>MANUAL METODOLÓGICO PARA LA GESTIÓN DE RIESGOS Y OPORTUNIDADES EN LA AGETIC</b>	
Código: UPAT/ADET-M03	Versión: 0	Aprobado: R.A. AGETIC/RA/0101/2022, de 12/08/2022

	de contabilidad es el tipo de datos <i>numeric</i> (o <i>decimal</i> ).
enum	El tipo de dato <i>enum</i> tiene las siguientes desventajas: se almacena en el catálogo del sistema, necesita <i>cast</i> , operadores numéricos y de cadena regulares no funcionan sobre él, no es portable, etc.
oid	El equipo central responsable de Postgres está eliminando gradualmente los OID. Los OID se llegarán a utilizar solo para referenciar a tablas del sistema.

#### Regla EBD14: Consideraciones sobre columnas numéricas

Todas las columnas que almacenen datos numéricos enteros deberán considerar su tamaño o longitud para luego seleccionar el tipo de datos adecuado. La siguiente tabla proporciona una relación:

Descripción del tamaño	Rango de valores	Tipo de datos adecuado
Si el entero x, es mayor o igual a 0 y menor o igual a 8 dígitos.	$x \geq 0$ y $x \leq 99999999$	int4
Si el entero x, es mayor a 8 dígitos y menor o igual a 16 dígitos.	$x > 99999999$ y $x \leq 9999999999999999$	int8
Si el entero x, es mayor a 16 dígitos.	$X > 9999999999999999$	numeric

Todas las columnas que almacenen datos numéricos enteros con decimales, deben ser definidas mediante los tipos de datos *numeric* y/o *decimal*, considerando que ambos son equivalentes.

#### Regla EBD15: Abreviaciones para identificadores

Utilizar las siguientes abreviaciones para identificadores de registro:

##### ID Identificador explícito

Se utilizará esta abreviación, cuando el valor de la columna se obtenga de un objeto de base de datos de tipo secuencia. No obstante, si la tabla no es transaccional el incremento puede generarse manualmente, o si la columna incluye letras y dígitos su valor podrá ser obtenido por una función tipo Hash.

	<b>MANUAL METODOLÓGICO PARA LA GESTIÓN DE RIESGOS Y OPORTUNIDADES EN LA AGETIC</b>	
Código: UPAT/ADET-M03	Versión: 0	Aprobado: R.A. AGETIC/RA/0101/2022, de 12/08/2022

Ejemplo:

*id*

### **ID Identificador implícito**

Se utilizará esta abreviación, cuando la columna utilice codificación para sus valores y las mismas están incluidas en tablas paramétricas.

Ejemplo:

*id\_estado*

### **Regla EBD16: Sufijos para columnas**

El presente manual no considera sufijos para la nominación de columnas.

### **Regla EBD17: Orden de las columnas**

Idealmente, las columnas deberán disponerse en el siguiente orden:

1. Columna llave primaria
2. Columnas de auditoría
3. Columnas de datos

### **Regla EBD18: Comentarios sobre columnas**

Toda tabla y sus campos deben incluir comentarios. Esos comentarios deberán reflejar de forma clara y en pocas palabras la utilidad de la tabla y los campos dentro de la misma. Estos comentarios podrán ser registrados al momento de la creación de la tabla, o en su defecto, posterior a ello.

## **10.5 Columnas de Auditoría**

### **Regla EBD19: Descripción de las columnas de auditoría**

Las columnas de auditoría describen el entorno temporal del registro. En orden de aparición, se sitúan después del campo identificador de la tabla (*id*) y antes de las columnas de datos:



## MANUAL METODOLÓGICO PARA LA GESTIÓN DE RIESGOS Y OPORTUNIDADES EN LA AGETIC

Código: UPAT/ADET-M03

Versión: 0

Aprobado: R.A. AGETIC/RA/0101/2022, de 12/08/2022

Orden	Nombre columna	Tipo de dato	Valor por defecto	Propósito
1	_estado	varchar (30) not null		Estado actual
2	_transaccion	varchar (30) not null		Última transacción (operación) ejecutada
3	_usuario_creacion	bigint not null	default 0	Autor creación
4	_fecha_creacion	timestamp not null	default now()	Fecha creación - fecha vigente desde
5	_usuario_modificacion	bigint not null	default 0	Autor último cambio
6	_fecha_modificacion	timestamp not null	default now()	Fecha último cambio - fecha vigente hasta

Ejemplo:

```
CREATE TABLE calendario (  
  -- columna LLAVE PRIMARIA  
  id serial8 not null,  
  -- columnas de AUDITORÍA  
  _estado varchar(30) not null,  
  _transaccion varchar(30) not null,  
  _usuario_creacion bigint not null,  
  _fecha_creacion timestamp not null,  
  _usuario_modificacion bigint not null,  
  _fecha_modificacion timestamp not null,  
  -- columnas de DATOS  
  tipo_prueba varchar(10) not null,  
  tipo_planificacion varchar(10) not null,  
  fecha_hora_inicio timestamp,  
  fecha_hora_fin timestamp,  
  id_estado varchar(10) not null,  
  id_etapa_area_grado bigint not null  
);
```

 <b>AGETIC</b> <small>Digitalizando Bolivia</small>	<b>MANUAL METODOLÓGICO PARA LA GESTIÓN DE RIESGOS Y OPORTUNIDADES EN LA AGETIC</b>	
Código: UPAT/ADET-M03	Versión: 0	Aprobado: R.A. AGETIC/RA/0101/2022, de 12/08/2022

## 10.6 Llaves y Restricciones

### Regla EBD20: Nominación de llaves primarias

Toda tabla debe tener una llave primaria, esta llave debe estar definida en lo posible sobre una única columna.

El nombre de la llave primaria será definido automáticamente por el ORM (Object-Relational Mapping) que utiliza la aplicación o por el gestor de base de datos.

Ejemplo:

```

Tabla:                empresas
Columna identificadora: id
Nombre resultante:    empresas_pkey

```

```

ALTER TABLE empresas
ADD PRIMARY KEY (id)
USING INDEX TABLESPACE idx_empresas;

```

Considerar que los actuales gestores de bases de datos crean por defecto índices únicos al momento de definir llaves primarias y *constraints* únicos, por tanto asegurar con alguna directiva del ORM que no se dupliquen estos índices.

### Regla EBD21: Llaves primarias identificadas con secuencias

En la medida de lo posible, las llaves primarias deben ser de un solo campo numérico con un secuencial asociado. Para tablas transaccionales o de gran volumen utilizar el tipo SERIAL8, para el resto solo SERIAL.

Ejemplo:

```

CREATE TABLE etapas_areas_grados (
    id          serial not null,
    ...
    PRIMARY KEY (id)
    USING INDEX TABLESPACE idx_registro
)
TABLESPACE dat_registro;

```

 <b>AGETIC</b> <small>Digitalizando Bolivia</small>	<b>MANUAL METODOLÓGICO PARA LA GESTIÓN DE RIESGOS Y OPORTUNIDADES EN LA AGETIC</b>	
Código: UPAT/ADET-M03	Versión: 0	Aprobado: R.A. AGETIC/RA/0101/2022, de 12/08/2022

### **Regla EBD22: Llaves primarias compuestas**

En la medida de lo posible, evitar el uso de llaves primarias compuestas. Si es el caso, el orden de las columnas de la llave primaria compuesta debe estar definido por el orden en las tablas de la relación actual. Es decir, empezando con la columna ID de la primera tabla, continuando con la columna ID de la segunda, y así sucesivamente.

Ejemplo:

Tabla: etapas\_areas\_grados  
Primera columna: id\_etapa  
Segunda columna: id\_area  
Tercera columna: id\_grado\_escolar  
Nombre resultante: etapas\_areas\_grados\_pkey

```
ALTER TABLE etapas_areas_grados
ADD PRIMARY KEY (id_etapa, id_area, id_grado_escolar);
```

### **Regla EBD23: Nominación de llaves únicas**

Similar a las llaves primarias, los nombres de las llaves únicas serán definidas por el ORM utilizado o por el gestor de base de datos.

Ejemplo:

Tabla: formularios  
Columna única: version\_form  
Nombre resultante: formularios\_version\_form\_key

```
ALTER TABLE formularios
ADD UNIQUE (version_form);
```

### **Regla EBD24: Nominación de llaves foráneas**

Similar a las llaves primarias, los nombres de las llaves foráneas serán definidas por el ORM o por el gestor de base de datos.

Ejemplo:



## MANUAL METODOLÓGICO PARA LA GESTIÓN DE RIESGOS Y OPORTUNIDADES EN LA AGETIC

Código: UPAT/ADET-M03

Versión: 0

Aprobado: R.A. AGETIC/RA/0101/2022, de 12/08/2022

Tabla: documentos  
Columna: id\_persona  
Tabla Referenciada: personas  
Nombre resultante: documentos\_id\_persona\_fkey

```
ALTER TABLE documentos  
ADD FOREIGN KEY (id_persona)  
REFERENCES personas (id);
```

### Regla EBD25: Nominación de restricciones de verificación (check)

Se recomienda el uso de *constraints* tipo check para el control de rango de valores que puedan tomar los datos a ser almacenados y recuperados.

La nominación de la restricción check estará formada por el nombre de la columna seguida del sufijo CHECK.

El formato es el siguiente:

*<NombreRestricciónCheck> = <NombreColumna>\_check*

Ejemplo:

```
CREATE TABLE empleados(  
id integer,  
nombre varchar(50),  
apellido varchar(50),  
genero varchar(10)  
constraint genero_check check(genero in ('FEMENINO','MASCULINO')),  
fecha_nacimiento timestamp  
);
```

## 10.7 Índices

### Regla EBD26: Nominación de índices de búsqueda

De manera similar a las llaves primarias, los nombres de índices serán definidos por el ORM o por el gestor de base de datos.

	<b>MANUAL METODOLÓGICO PARA LA GESTIÓN DE RIESGOS Y OPORTUNIDADES EN LA AGETIC</b>	
Código: UPAT/ADET-M03	Versión: 0	Aprobado: R.A. AGETIC/RA/0101/2022, de 12/08/2022

Ejemplo:

Tabla: contribuyentes  
Columna de búsqueda: nit  
Nombre resultante: contribuyentes\_nit\_idx

*CREATE INDEX ON contribuyentes (nit);*

### **Regla EBD27: Creación de índices en *tablespaces* para índices.**

Los índices deben crearse en *tablespaces* destinados específicamente para ello, que por lo general son distintos a los de tablas de datos. Seguir lo acordado en la regla de *tablespaces*.

Ejemplo:

Tabla: contribuyentes  
Tablespace de datos: dat\_empresas  
Tablespace de índices: idx\_empresas

*CREATE UNIQUE INDEX ON contribuyentes (id) TABLESPACE idx\_empresas;*

### **Regla EBD28: Asignación de índices sobre llaves primarias con USING INDEX**

Para asociar llaves primarias a índices existentes, debe hacerse uso de la sentencia USING INDEX. El índice debe existir previamente y ser de tipo UNIQUE.

Ejemplo:

Tabla: contribuyentes  
Índice existente: contribuyentes\_id\_pkey

*ALTER TABLE contribuyentes  
ADD PRIMARY KEY USING INDEX contribuyentes\_id\_pkey;*

## **10.8 Vistas**

### **Regla EBD29: Nominación de vistas**

	<b>MANUAL METODOLÓGICO PARA LA GESTIÓN DE RIESGOS Y OPORTUNIDADES EN LA AGETIC</b>	
Código: UPAT/ADET-M03	Versión: 0	Aprobado: R.A. AGETIC/RA/0101/2022, de 12/08/2022

La nominación de vistas comunes y vistas materializadas deberán seguir lo descrito en la regla de nominación de tablas. Es decir, serán tratadas como cualquier otra tabla convencional, con la sugerencia de llevar al inicio el prefijo VW.

El formato, podría ser el siguiente:

*<NombreVista> = vw\_<DescripciónVista>*

Ejemplo:

*CREATE VIEW vw\_empresas\_unipersonales AS ...*

## 10.9 Secuencias

### Regla EBD30: Nominación de secuencias

Como en otros casos, los nombres de secuencias serán definidos por el ORM o por el gestor de base de datos.

Ejemplo:

Tabla: obligaciones  
 Columna de secuencia: id  
 Nombre resultante: obligaciones\_id\_seq

```
CREATE TABLE obligaciones(
  id          serial8, -- valor por omisión: nextval('obligaciones_id_seq'::regclass)
  tipo       varchar(50),
  duracion   varchar(50),
  fecha_vigencia timestamp
);
```

## 10.10 PL/pgSQL

### Regla EBD31: Nominación de parámetros de entrada y salida

Para el caso de funciones y procedimientos, los parámetros de entrada (*input*) y salida (*output*) deberán comenzar con el prefijo “p\_”, debe ser escrito en minúscula y luego deberá ser sucedido por la descripción del parámetro.

 <p>AGETIC Digitalizando Bolivia</p>	<b>MANUAL METODOLÓGICO PARA LA GESTIÓN DE RIESGOS Y OPORTUNIDADES EN LA AGETIC</b>	
Código: UPAT/ADET-M03	Versión: 0	Aprobado: R.A. AGETIC/RA/0101/2022, de 12/08/2022

El formato es el siguiente:

*<NombreParámetro> = p\_<DescripciónParámetro>*

Ejemplo:

```
CREATE PROCEDURE proc_inicializar (p_parametro_entrada IN text)
AS $$
BEGIN
    RAISE NOTICE 'Parámetro de entrada: %', p_parametro_entrada;
END;
$$ language plpgsql;
```

### **Regla EBD32: Nominación en la declaración de variables básicas**

Toda variable de tipo de datos básica (ej.: *integer*, *varchar*, *numeric*, *timestamp*, etc.) que se declare en ámbito global o local, debe comenzar con el prefijo “v\_”, escrita en minúscula y luego sucedida por la descripción de la variable.

El formato es el siguiente:

*<NombreVariable> = v\_<DescripciónVariable>*

Ejemplo:

```
do $$
declare
    v_contador integer;
begin
    for v_contador in 1..5 loop
        raise notice 'valor contador: %', v_contador;
    end loop;
end; $$
```

### **Regla EBD33: Nominación en la declaración de variables tipo arreglo**

Toda variable de tipo arreglo base o arreglo compuesto debe empezar con el prefijo “a\_”, escrita en minúscula y luego sucedida por la descripción de la variable.

	<b>MANUAL METODOLÓGICO PARA LA GESTIÓN DE RIESGOS Y OPORTUNIDADES EN LA AGETIC</b>	
Código: UPAT/ADET-M03	Versión: 0	Aprobado: R.A. AGETIC/RA/0101/2022, de 12/08/2022

El formato es el siguiente:

*<NombreArreglo> = a\_<DescripciónVariable>*

Ejemplo:

```

create table personas (
  id integer,
  nombre text,
  apellido text
);

do $$
declare
  a_credito      integer[];
  a_lista_personas  personas[];
begin
  a_credito[1] := 99;
  a_lista_personas[1] := row(42, 'Juan', 'Pérez')::personas;
  raise notice 'Crédito: %', a_credito[1];
  raise notice 'Persona: %', a_lista_personas[1].nombre;
end; $$

```

#### **Regla EBD34: Nominación en la declaración de variables tipo *cursor* y *refcursor***

En cuanto a las variables de tipo cursor, estas deben comenzar con el prefijo “c\_” sucedida por la descripción de la variable.

El formato es el siguiente:

*<NombreCursor> = c\_<DescripciónVariable>*

Ejemplo:

```

do $$
declare
  r_cliente      record;
  c_clientes      cursor for select * from clientes order by nombre_cliente;

```



## MANUAL METODOLÓGICO PARA LA GESTIÓN DE RIESGOS Y OPORTUNIDADES EN LA AGETIC

Código: UPAT/ADET-M03

Versión: 0

Aprobado: R.A. AGETIC/RA/0101/2022, de 12/08/2022

```
begin
  for r_cliente in c_clientes loop
    raise notice 'Procesando %', r_cliente.nombre_cliente;
  end loop;
end; $$
```

### Regla EBD35: Nominación en la declaración de variables *record* y *rowtype*

En cuanto a las variables de tipo cursor, estas deben comenzar con el prefijo “r\_” sucedida por la descripción de la variable.

El formato es el siguiente:

*<NombreRegistro> = r\_<DescripciónVariable>*

Ejemplo:

```
do $$
declare
  r_cliente  clientes%rowtype;
  c_clientes cursor for select * from clientes order by nombre_cliente;
begin
  for r_cliente in c_clientes loop
    raise notice 'Procesando %', r_cliente.nombre_cliente;
  end loop;
end; $$
```

### Regla EBD36: Nominación de tipos definidos por el usuario

Todo tipo de dato definido por el programador deberá ser escrito en minúscula y deberá respetar la siguiente convención.

En el caso de arreglos utilizar el prefijo “t\_”:

*<NombreTipoArreglo> = t\_<DescripciónTipo>*

Ejemplo:

```
CREATE TYPE t_valores_array AS (
```



## MANUAL METODOLÓGICO PARA LA GESTIÓN DE RIESGOS Y OPORTUNIDADES EN LA AGETIC

Código: UPAT/ADET-M03

Versión: 0

Aprobado: R.A. AGETIC/RA/0101/2022, de 12/08/2022

```
    valor    varchar(50)[]
);

do $$
declare
    v_arreglo    t_valores_array;
begin
    v_arreglo.valor[0] = 'piedra';
    v_arreglo.valor[1] = 'papel';
    v_arreglo.valor[2] = 'tijera';
    raise notice 'len=%',array_length(v_arreglo.valor,1);
end; $$
```

En el caso de registros utilizar el prefijo “r\_”:

*<NombreTipoRegistro> = r\_<DescripciónTipo>*

Ejemplo:

```
CREATE TYPE r_clientes AS (
    id            integer,
    item          text,
    fecha_exp    timestamp without time zone,
    dias_rest    integer
);
```

### 10.11 Procedimientos y Funciones

#### Regla EBD37: Nominación de procedimientos

Todo procedimiento definido por el programador debe llevar el prefijo “proc\_”, escrito en minúscula y respetar la siguiente convención:

*<NombreProcedimiento> = proc\_<DescripciónProcedimiento>*

Los procedimientos tienen la forma:

```
CREATE PROCEDURE proc_<DescripciónProcedimiento> (
    p_<DescripciónParámetro> <IN/OUT> <TipoDato>, ... )
```



## MANUAL METODOLÓGICO PARA LA GESTIÓN DE RIESGOS Y OPORTUNIDADES EN LA AGETIC

Código: UPAT/ADET-M03

Versión: 0

Aprobado: R.A. AGETIC/RA/0101/2022, de 12/08/2022

```
AS $$  
  v_<DescripciónVariable> <TipoDato>;  
  ...  
BEGIN  
  ...  
EXCEPTION  
  WHEN others THEN  
  ...  
END;  
$$ language plpgsql;
```

Ejemplo:

```
CREATE PROCEDURE proc_inicializar (p_param IN text)  
AS $$  
BEGIN  
  raise notice 'Parámetro del procedimiento: %', p_param;  
END;  
$$ language plpgsql;
```

La descripción de los nombres de procedimientos debe contener por lo general verbos en infinitivo, y su longitud no debe exceder los 60 caracteres.

### Regla EBD38: Nominación de funciones

Toda función definida por el programador debe llevar el prefijo “func\_”, escrito en minúscula y respetar la siguiente convención:

*<NombreFunción> = func\_<DescripciónFunción>*

Las funciones tienen la forma:

```
CREATE FUNCTION func_<DescripciónFunción> (  
  p_<NombreParámetro> <IN/OUT> <TipoDato>, ... )  
RETURNS <TipoDato>  
AS $$  
  v_<DescripciónVariable> <TipoDato>;  
  ...  
BEGIN
```



## MANUAL METODOLÓGICO PARA LA GESTIÓN DE RIESGOS Y OPORTUNIDADES EN LA AGETIC

Código: UPAT/ADET-M03

Versión: 0

Aprobado: R.A. AGETIC/RA/0101/2022, de 12/08/2022

```
...  
EXCEPTION  
  WHEN others THEN  
...  
END;  
$$ language plpgsql;
```

Ejemplo:

```
CREATE FUNCTION func_incrementar (p_param IN integer)  
  RETURNS integer  
AS $$  
BEGIN  
  return p_param + 1;  
END;  
$$ language plpgsql;
```

La descripción de los nombres de funciones debe contener por lo general verbos en infinitivo, y su longitud no debe exceder los 60 caracteres.

### 10.12 Triggers

#### Regla EBD39: Nominación de disparadores (triggers)

Los disparadores de una tabla, ó también denominados triggers, tienen la forma:

*<NombreTrigger> = trg\_<DescripciónTrigger>*

Ejemplo:

Tabla: registro\_comercio  
NombreTrigger: trg\_generar\_matricula

```
CREATE TRIGGER trg_generar_matricula  
  BEFORE INSERT  
  ON registro_comercio  
  FOR EACH ROW  
  EXECUTE PROCEDURE func_generar_matricula();
```



## MANUAL METODOLÓGICO PARA LA GESTIÓN DE RIESGOS Y OPORTUNIDADES EN LA AGETIC

Código: UPAT/ADET-M03

Versión: 0

Aprobado: R.A. AGETIC/RA/0101/2022, de 12/08/2022

```
CREATE FUNCTION func_generar_matricula()
  RETURNS trigger
  language 'plpgsql'
AS $body$
DECLARE
  v_salida varchar(500);
BEGIN
  if (tg_op = 'INSERT' and new.arg_salida is null) then
    v_salida := 100;
    new.arg_salida := v_salida;
    new.id_estado := 'AC';
  end if;
  RETURN new;
END;
$body$;
```

El nombre del trigger debe tener un máximo de 60 caracteres, y una tabla podría tener más de un trigger asociado.

### **Regla EBD40: Los triggers no deben incluir código sofisticado**

El uso de triggers debe ser medido, con una implementación sencilla, y debe limitarse a invocar funcionalidades de la lógica del negocio mediante llamadas a funciones y procedimientos almacenados.

Ejemplo:

```
CREATE FUNCTION func_llamada_funcion_negocio()
  RETURNS trigger
  language 'plpgsql'
AS $body$
BEGIN
  if (tg_op = 'INSERT') then
    call func_sofisticada_negocio_A();
  elsif (tg_op = 'UPDATE') then
    call func_sofisticada_negocio_B();
  elsif (tg_op = 'DELETE') then
    call func_sofisticada_negocio_C();
  end if;
```



## MANUAL METODOLÓGICO PARA LA GESTIÓN DE RIESGOS Y OPORTUNIDADES EN LA AGETIC

Código: UPAT/ADET-M03

Versión: 0

Aprobado: R.A. AGETIC/RA/0101/2022, de 12/08/2022

```
    RETURN new;  
END;  
$body$;
```

### 10.13 Comentarios

En el presente contexto, los comentarios están destinados a insertar anotaciones legibles al programador en el código fuente de funciones y procedimientos.

#### **Regla EBD41: Comentarios sobre procedimientos y funciones**

Básicamente, lo que se busca es describir la funcionalidad del código.

Se ha de tener en cuenta que los comentarios necesitan mantenimiento igual que el código y, por tanto, que un comentario preciso y conciso es más fácil de mantener que uno largo, repetitivo y complicado.