

1. **OBJETIVO:**

El objetivo del presente documento de estándar y lineamientos es definir los criterios a seguir para la utilización de los motores de base de datos y nomenclatura de sus correspondientes objetos a ser usados por las aplicaciones desarrolladas por TUEMPRESA, que permitan identificar fácilmente a los objetos de base de datos, determinar a qué sistema pertenece, tipo de objeto y el tipo de información almacena, promover la colaboración entre los diferentes grupos de desarrollo permitiendo que objetos de base de datos creados para una aplicación se utilicen en el desarrollo de otras, facilitar el mantenimiento de aplicaciones haciendo que un módulo no esté ligado al programador del mismo.

2. **ALCANCE:**

El presente informe tiene como alcance su aplicación en los todos los ambientes de las bases de datos de TUEMPRESA.

3. **DOCUMENTOS DE REFERENCIA**

- Metodología de TUEMPRESA

4. **MARCO LEGAL**

- NO HAY

5. **DEFINICIONES**

Comité de Arquitectura: Es un cuerpo rector conformado por los arquitectos y especialistas de tecnología

6. **LINEAMIENTOS DE BASES DE DATOS:**

6.1. **Generales**

- a. La vigencia de los lineamientos y estándares debe tener una revisión anual, debido a las vicencias y nuevas tecnologías emergentes de las herramientas de gestión de base de datos.

6.2. **Responsabilidad**

- a. Es responsable del administrador de base de datos verificar la vigencia y soporte de la versión de los motores de bases de datos y su comunicación al responsable de la aplicación para su gestión correspondiente.
- b. Es responsabilidad del Gestor del Proyecto el cumplimiento del estándar para aplicaciones nuevas y mantenimiento que están su cargo.
- c. Es responsabilidad del administrador de base de datos, verificar y aprobar la aplicación del estándar de datos luego terminado las fases de análisis y diseño de una aplicación nueva, en un plazo no mayor a 5 días hábiles recibido la documentación.
- d. Es responsabilidad del comité de Arquitectura la revisión y mantenimiento del presente documento.

6.3. **Acceso y Seguridad**

- a. Crear y configurar usuarios de base de datos con privilegios restringidos a quien llamaremos **Usuario de Aplicación**, para el acceso de las aplicaciones con el esquema de base de datos.
- b. Se debe crear un solo usuario por cada tipo aplicación.
- c. Todas las cuentas son creadas por el Administrador de base de datos.
- d. La auditoría de la base de datos ORACLE es realizada utilizando el software propio de la base de datos, en cuanto a la auditoría a nivel aplicación debe ser implementada por el gestor de la aplicación.
- e.

- f. En ningún caso las cuentas de usuarios tendrán el privilegio de System Administrator (DBA).
- g. Se debe ofuscar o enmascarar los datos que muestren información sensible, solo para efectos de preparación de información para las pruebas correspondientes en los ambiente de certificación, previa autorización del propietario de la información y en coordinación con el oficial de seguridad de la información.
- h. La implementación del enmascaramiento está a cargo del administrador de base de datos.

6.4. Integridad de Data

- a. La integridad referencial será manejado a través de Constraints Primary Key y Foreign Keys de acuerdo al Modelo de Datos definido.
- b. En caso se requieran hacer validaciones sobre columnas se debe utilizar el constraint del tipo CHECK, salvo que se quiera hacer una validación de un USER DEFINED DATATYPE.
- c. Disponer del modelo conceptual, lógico y físico (Diagrama de E-R) del diseño de datos de cada proyecto, validados por el gestor y el comité de arquitectura en la etapa de análisis y diseño de cada proyecto.

6.5. Programación

- a. La implementación de Procedimientos Almacenados está permitido para realizar manipulación y procesamiento de altos volúmenes de información.
- b. No está permitido implementar lógica y reglas de negocio a nivel de base de datos.
- c. No se deben usar procedimientos encriptados. Salvo indicaciones y recomendaciones específicas de parte del oficial de seguridad en coordinación con el propietario de la información.
- d. Las vistas debe ser utilizadas para simplificar los queries.
- e. En ningún caso se otorgarán permisos sobre columnas, por lo tanto se debe crear una vista para que se le otorgue permiso de SELECT a toda la vista.
- f. Los tipos de datos definidos por usuario (User Defined Datatypes) debe ser debidamente documentados.

6.6. Conexiones al Servidor

Todos los sistemas a desarrollarse ~~se~~ deben de optimizar el uso de conexiones por usuario.

6.7. Performance y tuning de Base De Datos

- a. Solo para aplicaciones que sean necesarias se deben realizar un test con un mínimo de 100 conexiones concurrentes, pudiendo ser el test con una cantidad mayor de usuarios dependiendo del tipo de sistema, estas pruebas serán supervisadas por el DBA y el Gestor del Proyecto con el fin de garantizar el funcionamiento adecuado de la base de datos.
- b. Si el test no es aprobado satisfactoriamente, se debe corregir las consultas y en algunos casos el código de los sistemas.
- c. Antes del pase a Producción de una Base de Datos ésta debe ser analizada por el DBA en colaboración de los desarrolladores para optimizar al máximo el funcionamiento si es que la performance alcanzada no es adecuada en un plazo máximo de 2 días.

6.8. Despliegue en producción

Para que una Base de Datos pase a producción, Control de Calidad habrá gestionado que toda la documentación pertinente se encuentre registrada en la herramienta o repositorio respectivo, conteniendo el detalle de los siguientes puntos.

- a. **Requerimientos para el proyecto.-** Características Técnicas de Cliente y Servidores, software necesario y por ende licencias y tipo de licencias para el funcionamiento del sistema tanto a nivel Cliente como Servidor.

- b. **Descripción del proyecto.-** Breve resumen general del sistema, indicando áreas usuarias y relacionadas en el I/O del mismo.
- c. **Procedimientos de mantenimiento.-** Indicando ubicación de Fuentes y frecuencia de actualización y procedimiento del mismo, así como los instaladores del aplicativo, los cuales debe estar en directorio independiente y compartido para cualquier necesidad de reinstalación en el cliente, igualmente procedimiento de reinstalación de la parte de Servidor.
- d. **Base Datos.-** Tablas, relaciones entre ellas, índices, cantidad de tablas, tablas interfaces, cómo es el flujo de la información. Si existen archivos planos que se trasladan de otros servidores al servidor del Sistema (si fuese así nombre, teléfono de los que mandan el archivo plano). En resumen diseño completo de la Base de Datos así como de las interfaces, incluyendo scripts de generación de índices y constraints respetando el orden de dependencia, para los casos de restauración y/o mantenimiento de la Base de Datos.
- e. **Diccionario de Datos.-** Al momento de adjuntar los script de las tablas, estos debe incluir los comentarios correspondientes de todos los objetos creados en comandos SQL.
- f. **Proyección de crecimiento de la información.-** Proyección anual, mensual y en línea de la información para determinar las necesidades de espacio en disco, asimismo, proceso de cierre periódico para el guardado y limpieza de tablas cuya información histórica podría ser extraída de la Base de Datos para la recuperación de espacio de Disco, igualmente para el caso de archivos de importación o exportación (interfaces con otros sistemas), indicando tamaños y tiempo de vida en disco, los criterios deben ser definidos en coordinación con el usuario responsable del aplicativo y el comité de arquitectura.
- g. **Conexiones con otras bases de datos.-** La conexión a emplearse debe ser por jdni.xml.
- h. **Personal relacionado al proyecto.-** Personas que ingresan al sistema, persona de aplicativos responsable del sistema, otras personas relacionadas con el proyecto, se necesita nombres, teléfonos, ubicación.
- i. **Procedimiento de creación de nuevos usuarios en el Sistema.-** Se especifica el área responsable de la misma y login que utilizan para la conexión a la Base de Datos.
- j. **Formato de Pase a Producción.-** Para realizar el pase a producción se solicitará que se adjunte el formato de acuerdo a la metodología "INICIALES DE TU METODOLOGIA".
- k. **Archivo de script.-** Los archivos de scripts SQL tiene que seguir las siguientes pautas y recomendaciones:
 - Se tienen que enumerar los archivos con el número correspondiente al orden de ejecución de la instalación (ejemplo: *01-tablas.sql*, *02-vistas.sql*, *03-procedimientos.sql*).
 - Los nombres de archivo no tienen que incluir espacios en blanco ni caracteres especiales.
 - Evitar utilizar sentencias DDL y DML en el mismo archivo, utilizar por orden de ejecución como prioridad a los DDL seguido de los DML.
 - Se ha de evitar enviar archivos muy extensos, ya que si falla algún punto del archivo, éste se continúa ejecutando hasta el final, y puede hacer que la vuelta atrás de la instalación sea muy complicada.
 - No se tienen que enviar los scripts DML con sentencias commit. Si la ejecución del script de carga es correcta, los administradores de la base de datos ya harán el commit correspondiente. Esto puede evitar muchos problemas ya que si se envían cinco scripts y todos hacen commit, pero el último de ellos falla, al haberse hecho commit en los scripts anteriores no es posible volver atrás la instalación completa haciendo un rollback.

6.9. Excepciones

Los ítems mencionados en éste numeral, son considerados excepciones, las cuales no están definidas en el Numeral 6.2 "Arquitectura de Base de Datos":

- a. Se crearán usuario de consulta a la base de datos con la aprobación del propietario de la información y el oficial de seguridad con accesos restringidos de solo lectura.

- b. Una solución personalizada propuesta por el proveedor, compatible con la base de datos ORACLE, pero que no trabaja con el sistema operativo especificado en el presente documento.
- c. La solución propuesta por el proveedor, la cual trabaja en una determinada plataforma de base de datos que no es ORACLE Database, Ejm (SQL SERVER, MYSQL, etc).
- d. La solución personalizada propuesta por el proveedor, que utiliza la base de datos ORACLE, pero que su modelo de datos no pueda adecuarse a la nomenclatura específica en el presente documento.
- e. Se otorgarán permisos sobre columnas, con fines de seguridad para datos sensibles previa autorización del propietario de la información y el oficial de seguridad.

Finalmente, para todas las excepciones se solicita la aprobación del Comité de Cambios de Ingeniería y Arquitectura, con el fin de garantizar que el software de base de datos propuesto por el proveedor pueda ser soportado por los responsables de las administración de la base de datos y que se disponga de la infraestructura adecuada para su implementación, esta aprobación debe evidenciarse en un acta de conformidad, la cual debe ser presentada por el Gestor del Proyecto al momento de iniciar la ejecución del proyecto (como parte de los entregables finales de la planificación del proyecto).

7. ESTÁNDAR DE BASE DE DATOS

7.1. Consideracion generales

- a. Los nombres de los objetos de base de datos será como máximo de 30 caracteres y en mayúsculas, y sólo pueden incluir los caracteres A-Z, 0-9 y separando cada palabra entre guion bajo (_).
- b. No usar abreviaciones, salvo el nombre exceda los 30 caracteres.
- c. Se debe registrar metadata a la base de datos, agregando comentarios a las tablas y campos, sobre todo a los booleanos.
- d. Dentro de la programación lógica de los programas no deben referenciar a los propietarios de los objetos, al momento de invocar tablas, funciones, procedimientos, vistas, etc., para ello deben usar sinónimos si es requerido.
- e. Las letras acentuadas se reemplazarán con las equivalentes no acentuadas, y en lugar de la letra eñe (Ñ) se utilizará (NI). Ejm: ANIO_EXPEDIENTE.
- f. El juego de caracteres de las bases de datos es WE8ISO8859P15 (NLS_CHARACTERSET = WE8ISO8859P15).
- g. El National Character set es AL16UTF16 (NLS_NCHAR_CHARACTERSET = AL16UTF16).

7.2. Arquitectura de Base de Datos

La Arquitectura de la base de datos en todos los ambientes de Producción, Certificación y Desarrollo es de 64 Bits.

A continuación se detalla la matriz de compatibilidad para trabajar con la base de datos ORACLE:

BASE DE DATOS	SISTEMA OPERATIVO
Oracle Database 12c R1	IBM (64-bit)
	Linux x86-64 Red Hat Enterprise

El motor de la base de datos debe ser:

- Oracle Database - Enterprise Edition 12c, para la realización de nuevas aplicaciones.
- Para el caso de mantenimiento de aplicaciones preexistentes las versiones disponibles de la base de datos Oracle 9i y Oracle 11g R2, según corresponda.

7.3. Nomenclatura general para objetos en sistemas transaccionales

Los objetos son estructuras lógicas que se almacenan dentro de la base de datos, forman parte de los procesos de las Aplicaciones Informáticas, por lo que se ha definido el siguiente estándar:

<IdAplicación(a)>_<IdObjeto(b)>_<NombreObjeto(SegúnNomenclatura)(c)>

Donde:

- a. Identificador de Aplicación.-** Las aplicaciones son identificados a través de un código de identificación de 3 (tres) caracteres asignados por el responsable del inventario de aplicaciones de TUEMPRESA. Este identificador formará parte de la nomenclatura de los objeto para identificar a que sistema pertenece. A continuación se indican algunos identificadores de Sistema existentes.

Ejemplo:

Identificador de Aplicación	Nombre de Aplicación
PVO	Plataforma Virtual TUEMPRESA
SGD	Sistema General Documentario
SGTD	Intranet de la nueva plataforma de revisión
MDI	Modelo de Datos Institucional

- b. Identificador de Objeto.-** Corresponde al tipo de objeto (tabla, vista, procedimiento, paquete, secuencia, java, trigger, etc.) al que pertenece el objeto e información adicional del objeto según sea el caso, tal como se muestra en el siguiente cuadro.

Ejemplo:

Objeto	Identificador de Objetos
Tabla	T<Tipo Contenido>
Vista	V<Tipo Contenido>
Procedimiento	PRC
Paquete	PQT
Función	FUN
Trigger	TRG
Secuencia	SEQ
Data Base Link	LNK
Índice	IX<Tipo Indice>

- c. Nombre de Objeto.-** Es el nombre del objeto abreviado; para nombres compuestos irá separado por el carácter subrayado '_ '.

Ejemplo:

Tipo de Objeto	Nombre de Objetos
Tabla	SGD_TC_EXPEDIENTES, SGD_TD_DOCEXPEDIENTES, SGD_TC_ARCHIVOS,

	SGD_TM_TIPOTRAMITE MDI_TM_CLIENTES
Vista	SGD_VC_EXPEDIENTES, SGD_VD_DOCEXPEDIENTES, SGD_VM_TIPOTRAMITE SGD_VQ_REPORTES
Procedimiento Almacenado	SGD_PRC_CREAR_EXPEDIENTE SGD_PRC_GUARDAR_PAGO
Función	SGD_FUN_DIAS_HABILES SGD_FUN_CALCULA_SANCION
Paquetes	SGD_PQT_ADM_SGD SGD_SGD_PQT_MDI
Trigger	SGD_TRG_ENVIO_PUBLICACION
Secuencia	SGD_SEQ_PROCESOS

7.3.1. Esquema y Usuarios

- La longitud máxima de este nombre es de 15 caracteres.
- Debe ser un nombre único en la base de datos que identifique al aplicativo, el cual será coordinado con el Administrador de Base de Datos de TUEMPRESA.

Nomenclatura: <Prefijo>_<Nombre>

Ejemplo:

Prefijo	Nomenclatura	Detalle
ES	ES_SGTD	Esquema (Propietario)
US	US_SGTD_REST US_SGTD_WEB US_SGTD_WSOAP	Usuario de Aplicación
UC	UC_DGSS UC_JQUISPE	Usuario de consulta

7.3.2. Tabla

- La nomenclatura estándar para la definición de tablas de datos para todos los procesos y aplicaciones de TUEMPRESA, en sus distintas plataformas y motores de base de datos, debe seguir un formato el cual dependerá de distintos factores: Nombre de Aplicación y Nombre de la Tabla.
- El prefijo de la aplicación constará de 3 caracteres, el cual será coordinado con el Administrador de Base de Datos de TUEMPRESA.
- Debe usar el guion bajo (_) entre el prefijo y el nombre de la tabla.
- Debe ser un nombre referente a la Entidad a Modelar.
- El nombre debe estar en singular, no en plural.
- Cada tabla debe tener un comentario descriptivo para ser registrado en la base de datos utilizando el comando COMMENT ON TABLE, el cual permite introducir comentarios a las tablas.

Formato:

[Prefijo de Aplicación]_[Identificador de objeto]_[Nombre de la Tabla]

Ejemplo: SGTD_TD_PAGO_LEGAL

En donde:

- Prefijo de Aplicación: **SGTD**

- Identificador de objeto: **T<Tipo de Contenido>**
- Nombre de Tabla: PAGO_LEGAL (Tabla de bases legales)

Tipo Contenido	Abreviatura (1 carácter)	Identificador de Objeto
Maestra	M	TM
Trazabilidad (Movimiento)	Z	TZ
Cabecera	C	TC
Detalle	D	TD
Auxiliar	X	TX
Log	L	TL
Histórica	H	TH
Temporal	T	TT
Parámetros	P	TP
Auditoria	A	TA

7.3.3. Columna

- a. El nombre debe estar compuesta por la combinación de un prefijo(1) y un nombre(2)

Prefijo (1)	Nombre (2)
ID	UBIGEO
CO	ADMINISTRADO
NU	EXPEDIENTE

Donde:

Prefijo, define el contenido de la columna, no confundir con el tipo de dato de la columna. Tiene una longitud de 2 caracteres.

Prefijo	Descripción	Tipo Dato
ID	Identificador de llave primaria	Char
CO	Código	Varchar / Numeric
DE	Descripción	Varchar
CM	Comentario	Varchar
NU	Número	Numeric
SE	Secuencia	Numeric
IN	Indicador	Char
TI	Tipo	Char
ES	Estado	Char
FE	Fecha/Hora	Date
MO	Monto	Decimal
PC	Porcentaje	Decimal
CA	Cantidad	Numeric/ Decimal/ Integer
NO	Nombre	Varchar
AP	Apellido	Varchar
ME	Medida	Numeric / Decimal
US	Usuario	Varcha / Numérico
TE	Terminal (Host)	Varcha / Numérico
DI	Dirección	Varchar

IP	dirección IP	Varchar / Char
PW	Password	Varchar
ED	Edad	Varchar / Char
FL	Flag	Char

- b. Los nombres de las columnas deben describir el dato que van a almacenar de manera entendible; si se usan abreviaturas (porque se excedió del tamaño límite de 30 caracteres), éstas debe ser mnemotécnicas para facilitar la programación de los Stored Procedures y otros elementos.
- c. Para los nombres en las columnas de cada tabla debe ser separadas cada palabra por el guión bajo (_) y tan solo tienen la limitación del tamaño máximo de 30 caracteres.
- d. Para los campos clave (PK) deben ubicarse al inicio de la definición de la tabla (deben ser los primeros), además el nombre del campo clave debe estar compuesto por "ID_" + nombre de la tabla en singular (para claves no compuestas).
Ejm: Tabla Localidades => ID_LOCALIDAD.
- e. Para los campos de clave foránea es similar al de la clave primaria, y si hubieran más de 2 campos de tipo clave foránea referidos a la misma tabla, entonces aumentar al medio un nombre más que complementa el dato.
- f. Los campos de tipo Boolean deben nombrarse con una palabra que indique su estado en verdadero.
- g. Los campos de tipo Fecha y hora deben nombrarse con una palabra que indique si es Fecha y/u hora.
- h. Las tablas de mayor relevancia como la Maestra, trazabilidad, cabecera, detalle y auditoria deben incluir los campos de auditoría de la creación y actualización de los registros. Estos campos de auditoría NO deben ser creados para aquellas tablas que contengan data de manera temporal, logs, históricas y otras que no sea relevante.
La siguiente tabla especifica las características de los campos de auditoría:

Nombre del campo	Tipo	Descripción	Ejemplo	Valor NULL
US_CREACION	VARCHAR2(38)	Usuario creador	XQUISPE	Not NULL
IPTE_CREACION	VARCHAR2(38)	Terminal de creación	192.168.1.1	Not NULL
FE_CREACION	DATE	Fecha y hora de creación	18/09/2012 10:43:01 a.m.	Not NULL
US_ACTUALIZACION	VARCHAR2(38)	Usuario modificador	USU_SGD	NULL
IPTE_ACTUALIZACION	VARCHAR2(38)	Terminal de modificación	192.168.1.1	NULL
FE_ACTUALIZACION	DATE	Fecha y hora de modificación	20/09/2012 11:13:10 a.m.	NULL

i. Cada campo debe tener un comentario descriptivo para ser registrado en la metadata de la base de datos utilizando el comando **COMMENT ON COLUMN**. Además se debe tener presente que:

- Cuando el campo de la tabla es un dato confidencial, es necesario que se especifique para que quede registrado en la Base de Datos.
Ejm: **COMMENT ON COLUMN** ES_SGTD.SGTD_TD_DJURADA.CO_DJURADA **IS** 'Código de la declaración jurada. Dato (CONFIDENCIAL).';
- Cuando se trate de un campo Primary key, y si este utiliza una secuencia, en el comentario debe indicar el nombre de la secuencia.
Ejm: **COMMENT ON COLUMN** ES_SGTD.SGTD_TC_ACTA_ID_ACTA **IS** 'Identificador del registro en la tabla de Actas de Control de Calidad. Secuencia: SGTD_SEQ_ACTA';
- Cuando se trate de un campo identificador de llave foránea (Foreign Key), en el comentario debe indicar la tabla a la que tiene referencia.
Ejm: **COMMENT ON COLUMN** ES_SGTD.SGTD_TM_ACTVISITA.ID_UNIDAD_FISCAL **IS** 'Foránea de la tabla Unidad Fiscalizadora';
- Cuando se trate un campo del tipo flag o booleano, se debe especificar los posibles valores que contendrá.
Ejm: **COMMENT ON COLUMN** ES_SGTD.SGTD_TD_ALMACENAMIENTO.ES_ALMACEN **IS** 'Flag indicador si el registro se encuentra Activo = 1 o No_Activo = 0';

7.3.4. Tablas de auditoria

- La nomenclatura estándar para la definición de tablas de auditoria es definido en el punto 6.3.2. a quien se debe asignarle el prefijo TA de (Tabla de Auditoria).
- El nombre debe estar en singular, no en plural.
- Cada tabla debe tener un comentario descriptivo para ser registrado en la base de datos utilizando el comando **COMMENT ON TABLE**, el cual permite introducir comentarios a las tablas.
- Los campos de auditoria adicionales a las ya existentes en las tablas de datos deben contener los siguientes campos.

Nombre del campo	Tipo	Descripción	Ejemplo	Valor NULL
TI_ACCION	VARCHAR2(3)	INS:INSERT; UPD:UPDATE; DEL: DELETE;	INS	Not NULL

- e. De la misma manera se adiciona los comentarios correspondientes
Ejm: **COMMENT ON COLUMN ES_SGTD.SGTD_TA_EXPEDIENTE.TI_ACCION IS**
'Indica el tipo de acción realizada en el registro INS: INSERT UPD: UPDATE DEL: DELETE';

7.3.5. Constraint

La forma de nombrar un Constraint dependerá de distintos factores: Nombre de Aplicación, Nombre del Constraint y Tipo de Constraint. Cada palabra debe estar separada por el guión bajo (_). La estructura del nombre de un Constraint es la siguiente:

Formato:

[Sufijo del tipo de Constraint]_[Prefijo aplicación]_[Nombre del Constraint]

Donde:

Tipo de Constraint	Sufijo
Primary Key	PK
Foreign Key	FK
Unique	UK
Check Constraint	CK

Ejemplo: **PK_SGTD_TD_ACTIVIDAD**

En donde:

Tipo de constraint: **PK** (Primary Key)

Prefijo aplicativo: **SGTD**

Nombre del Constraint: **ACTIVIDAD**

7.3.6. Índice

- La forma de poner el nombre a un índice dependerá de distintos factores:
- Nombre de la tabla y el Nombre del campo a indexar.
- Si se trata de un índice compuesto por más de una columna indicar un nombre que permita describir al índice creado

Formato:

[Nombre de Tabla]_[Nombre del Índice]_[Sufijo de Índice]

Sufijo	Nombre
IDX	Índices

Ejemplo: **IDX_MDI_CLIENTE_AP_MATERNO**

En donde:

- Prefijo de índice: **IDX**
- Nombre de la tabla: **MDI_CLIENTE**
- Nombre del campo: **AP_MATERNO**

7.3.7. Vista y Vista Materializada

- La forma de nombrar una vista, sigue el mismo formato, como se nombra a una tabla.
- Se debe colocar nombres descriptivos ya que una vista se puede utilizar para devolver el valor de varias tablas.
- La forma de poner el nombre a una vista dependerá de distintos factores: Nombre de Aplicación, Nombre de la Tabla, Objeto de base de datos.

Formato:

[Prefijo de Aplicación]_[Nombre del objeto]_[Sufijo objeto de base de datos]

Objeto de Base de datos	Sufijo
Vista	VW
Vista Materializada	VM

Ejemplo: SIGA_VM_PLANILLA

En donde:

- Nombre de Aplicación: SIGA
- Prefijo Objeto de base de datos: VW
- Nombre de la Vista: PLANILLA (Tabla de Planilla)

7.3.8. Synonyms

Los sinónimos normalmente se usarán para ocultar el propietario, la ubicación o el nombre real de una tabla (así otros usuarios la pueden acceder sin importar quien la haya creado o en donde se encuentre), también es utilizado para proporcionar a los usuarios nombres de objetos menos complicados que los reales.

7.3.9. Package, Procedure, Function, Sequence y Trigger

- Debe empezar con el prefijo identificador de la aplicación, seguida por el identificador de objeto y terminando con el nombre del objeto.

Formato:

<IdAplicación(1)>_<IdObjeto(2)>_<NombreObjeto(SegúnNomenclatura)(3)>

Ejemplo:

Tipo de Objeto	Sufijo	Nombre de Objetos
Paquetes	PKG	SGD_PQT_ADM_SGD SGD_PQT_MDI
Procedure	PRC	SGD_PRC_CREAR_EXPEDIENTE SGD_PRC_GUARDAR_PAGO
Function	FUN	SGD_FUN_DIAS_HABILES SGD_FUN_CALCULA_SANCION
Trigger	TRG	SGD_TRG_ENVIO_PUBLICACION
Sequence	SEQ	SGD_SEQ_PROCESOS

- Para todos los objetos Package, Procedure, Function y Triggers incluir antes del BEGIN comentarios del propósito del objeto creado, en caso el objeto sea modificado debe indicarse un comentario en la sección de revisiones, tal como se muestra en el siguiente Ejemplo:

CREATE FUNCTION SGTD_FUN_DIAS_HABILES (WFINI DATE, WFFIN DATE)

```

RETURN NUMBER IS
/*****
NOMBRE:   SGTD_FUN_DIAS_HABILES
PROPOSITO: Devuelve el número de días hábiles (Lunes a Viernes) entre dos fechas dadas
REVISIONES:
    Fecha      Autor      Descripción
-----
DD/MM/YYYY  Juan Quispe  1. Creador del procedimiento.

NOTAS:
Incluir comentarios que brinden valor y seguimiento en el objeto.

*****/
BEGIN
    Cuerpo de la función
    RETURN (XXX);
END;
/

```

- c. Para todos los objetos Package, Procedure, Function y Triggers, cada vez que se requiera su actualización se debe colocar el identificador de inicio y fin del cambio. Inicio del Cambio - @001i, Fin del Cambio - @001f

Tal como lo muestra el siguiente ejemplo:

```

/*@001i
SELECT DNNCNTE_SEQ.NEXTVAL INTO v_IDDENUNCIANTE FROM DUAL;
INSERT INTO
SGTD_DENUNCIANTE(NOMBRE,DIRECCION,DOCUMENTO,CORREO,TELEFONO,OBSERVACIONES,UBIGEO_ID,
D,D ENUNC_ID, ID)
VALUES
(v_NOMBRE,v_DIRECCION,v_DOCUMENTO,v_CORREO,v_TELEFONO,v_OBSERVACIONES,v_UBIGEO_ID,v_ID
DENUNCIA,v_IDDENUNCIANTE);
/*@001f */

```

7.3.10. Roles

La forma de poner el nombre a un Rol dependerá de distintos factores:

Nombre de Aplicación, Nombre del Rol, Tipo del Rol que se asignará al grupo de permisos de los procesos y aplicaciones.

Formato:

[Prefijo aplicación]_[Nombre del Rol]_[Sufijo de Objeto]

Objeto de Base de datos	Sufijo
Roles	ROL

Ejemplo: SGTD_ROL_ADMIN

En donde:

Prefijo aplicación: SGTD

Prefijo de Objeto: ROL

Nombre del Rol: ADMIN

7.4. Nomenclatura para la creación de objetos en inteligencia de negocios

La nomenclatura necesaria para la definición de los componentes y variables que serán declarados y/o usados durante parte de integración y carga de los datos en las estructuras del Data Warehouse (ETL) tendrán en esencia las mismas reglas de la base de datos transaccional, con las diferencias siguientes:

7.4.1. Prefijo para todos los objetos Warehouse

El prefijo constará de un mínimo de 3 caracteres, el cual será coordinado con el Administrador de Base de Datos de TUEMPRESA.

Después de la identificación de la aplicación seguido por guión bajo se identifica el objeto de acuerdo a la siguiente tabla:

Formato: [Prefijo DW]_[Tipo de Objeto]_[Nombre del Objeto]

Tipo de Objeto	Significado del Objeto	Equivalencia
PQ	Paquete	Workflow
FD	Flujo de Datos	Dataflow
GP	Grupo de Paquetes	Workflows
FP	Flujo de Proceso	Job
FR	Flujo de Reproceso	Job
DS	Conexión	Datastore
AQ	Componente de Arquitectura	----

Ejemplo: XYZ_PQ_Actividad

En donde:

Prefijo DW: XYZ (Prefijo que Identifica a la aplicación que pertenece) Tipo de Objeto: PQ

Nombre del Objeto: Actividad

7.4.2. Variables y parámetros

Para las variables utilizadas en los procesos no pueden tener prefijo por restricciones de la herramienta, para estas variables el estándar será:

Formato:

[\$][Tipo de Objeto]_[Nombre del Objeto]

Tipo de Objeto	Significado del Objeto	Equivalencia
VG	Variable Global	Global Variable
VL	Variable Local	Local Variable
P	Parámetro	Parameter

- a. **Variable Global:** La declaración de la variable global estará compuesta por la concatenación del símbolo \$, el cual es necesario para la declaración de variables y/o parámetros, el termino **VG** (iniciales de Variable Global), y el **nombre de la variable global** que se desee asignar, si este nombre está compuesto por muchas palabras, se usara la primera letra de cada palabra en Mayúscula para unir el conjunto de palabras.

Para el mejor entendimiento se separara con un **guion bajo** (_) el término VG y el nombre de la variable.

Ejemplo: \$VG_CodigoRegistro

En donde:

- Prefijo: \$
- Tipo de Objeto: VG
- Nombre del Objeto: CodigoRegistro

- b. **Variable Local:** La declaración de la variable local estará compuesta por la concatenación del símbolo \$, el cual es necesario para la declaración de variables y/o parámetros, el termino **VL** (iniciales de Variable local), y el **nombre de la variable local** que se desee asignar, si este nombre está compuesto por muchas palabras, se usara la primera letra de cada palabra en Mayúscula para unir el conjunto de palabras.

Para el mejor entendimiento se separara con un **guion bajo** (_) el término VL y el nombre de la variable.

Ejemplo: \$VL_NombreFlujoDato

En donde:

- Prefijo: \$
- Tipo de Objeto: VL
- Nombre del Objeto: NombreFlujoDato

- c. **Parámetro:** La declaración de un parámetro estará compuesta por la concatenación del símbolo \$, el cual es necesario para la declaración de variables y/o parámetros, el termino **P** (iniciales de Parámetro), y el **nombre del parámetro** que se desee asignar, si este nombre está compuesto por muchas palabras, se usara la primera letra de cada palabra en Mayúscula para unir el conjunto de palabras.

Para el mejor entendimiento se separara con un **guion bajo** (_) el término P y el nombre de la variable.

Ejemplo: \$P_ParametroEntrada

En donde:

- Prefijo: \$
- Tipo de Objeto: P
- Nombre del Objeto: ParametroEntrada

7.4.3. Modelamiento Dimensional

Un modelo de datos dimensional contiene esquemas Estrella en su diseño, los mismos que están compuestos por tablas de Hechos y tablas Dimensionales. Además, existen las tablas Data Warehouse, como una fuente de información de datos transformados y separados físicamente de la aplicación donde se encuentran los datos del ambiente operacional. A continuación presentamos la nomenclatura para estos tres tipos de tablas:

Formato:

[Prefijo DW]_[Sufijo]_[Nombre del Objeto]

Nombre

Sufijo

Data Warehouse	DW
Hechos	HE
Dimensión	DM

Ejemplo: XYZ_DW_UNIDAD_OPERATIVA

En donde:

- Prefijo DW: XYZ (Prefijo que Identifica a la aplicación que pertenece)
- Sufijo: DW
- Nombre del Objeto: UNIDAD_OPERATIVA