

SAT

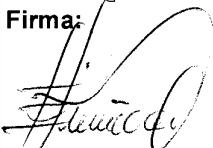
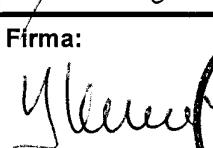
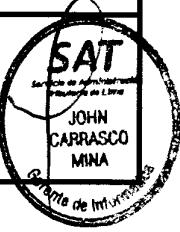
*Servicio de Administración
Tributaria de Lima*

ESTÁNDARES PARA EL DESARROLLO DE SISTEMAS EN SQL

Documento Técnico: GIN-AGS-DT003

Versión: 01

GERENCIA DE INFORMÁTICA

Elaborado por: Edgard Márquez Aylas Cargo: Especialista de Gestión de Soluciones de TI I Fecha: 06/03/2014	Firma: 
Revisado por: Duglas Quiñones Casas Cargo: Especialista de Gestión de Soluciones de TI II Fecha: 14/05/2014	Firma: 
Aprobado por: Jhon Carrasco Mina Cargo: Gerente de Informática Fecha: 15/05/2014	Firma:   JOHN CARRASCO MINA Gerente de Informática

SAT Servicio de Administración Tributaria de Lima	Tipo: DOCUMENTO TÉCNICO	Código: GIN-AGS-DT003 Versión: 01
	Título: ESTÁNDARES PARA EL DESARROLLO DE SISTEMAS EN SQL	Fecha de Vigencia: 16/05/2014
		Página: 2 de 37

HOJA DE CONTROL DE CAMBIOS

ÍTEM	TEXTO MODIFICADO	CÓDIGO-VERSIÓN	FECHA	RESPONSABLE
1	Elaboración inicial del documento	GINDDSD T007 – V.1	30/11/2009	División de Desarrollo de Sistemas
2	Elaboración de la Segunda Versión	GINDDSD T007 – V.2	28/08/2012	
3	Cambio del código del documento por el nuevo Manual Interno			
4	En la sección 1.1.1 se adicionó un nuevo tipo de dato "XML"			
5	En la sección 1.2 se adicionó 3 palabras claves			
6	En la sección 1.2.3 se precisa el nombre de las tablas dimensionales, se modifica la tabla de tipo, se adiciona más acrónimos a la tabla, se precisa el nombre de las tablas extendidas y ejemplos de nombres de las tablas dimensionales.			
7	Se agregaron la secciones: 1.2.4 Nombre de Tablas Temporales Físicas 1.2.5 Nombre de Tablas Temporales Físicas para Migración 1.2.19 Nombre de Paquetes 1.2.20 Nombre de Tareas (Job) 1.2.21 Nombre de Inicio de Sesión 1.2.22 Nombre de Objetos de Baja 2.4 Campos Adicionales para Auditoría 2.5 Consideraciones del uso de los campos de auditoría 4.3 Anexo 3: Modelo de Datos del SIAT	GINAGSD T003 – V.1	06/03/2014	A.F. Gestión de Soluciones de TI
8	En la sección 1.5 se adicionaron nuevos códigos de reglas.			
9	Se modificaron las secciones: 1.2.1 Nombre de Tablas 1.2.17 Nombres de Variables 1.3 Convenciones de Código 2.1 Tablas Auditables con Histórico 2.2 Tablas Auditables sin Histórico 2.3 Tablas Auditables por Seguridad			



SAT <i>Servicio de Administración Tributaria de Lima</i>	Tipo: DOCUMENTO TÉCNICO	Código: GIN-AGS-DT003 Versión: 01
	Título: ESTÁNDARES PARA EL DESARROLLO DE SISTEMAS EN SQL	Fecha de Vigencia: 16/05/2014
		Página: 3 de 37

ÍNDICE

1. ESTÁNDARES DE TRANSACT-SQL	4
1.1. NOMENCLATURA PARA LOS ELEMENTOS DE PROGRAMACIÓN	4
1.1.1. PREFIJOS DE TIPOS DE DATOS	4
1.2. DECLARACIONES	6
1.2.1. NOMBRE DE BASES DE DATOS.....	8
1.2.2. NOMBRE DE ESQUEMAS	8
1.2.3. NOMBRE DE TABLAS	9
1.2.4. NOMBRE DE TABLAS TEMPORALES FÍSICAS.....	11
1.2.5. NOMBRE DE TABLAS TEMPORALES FÍSICAS PARA MIGRACIÓN	11
1.2.6. NOMBRE DE COLUMNAS	12
1.2.7. NOMBRE DE CONSTRAINTS (PRIMARY KEY)	13
1.2.8. NOMBRE DE CONSTRAINTS (FOREIGN KEY)	13
1.2.9. NOMBRE DE CONSTRAINTS (DEFAULT)	13
1.2.10. NOMBRE DE CONSTRAINTS (CHECK)	14
1.2.11. NOMBRE DE ÍNDICES	14
1.2.12. NOMBRE DE VISTAS	14
1.2.13. NOMBRE DE PROCEDIMIENTOS ALMACENADOS	14
1.2.14. NOMBRES DE FUNCIONES DEFINIDAS POR EL USUARIO.....	15
1.2.15. NOMBRES DE TRIGGERS.....	16
1.2.16. NOMBRES DE LOS PARÁMETROS	16
1.2.17. NOMBRES DE VARIABLES	17
1.2.18. NOMBRES DE ARCHIVOS XML	17
1.2.19. NOMBRES DE PAQUETES.....	17
1.2.20. NOMBRES DE TAREAS (JOB).....	17
1.2.21. NOMBRES DE INICIO DE SESIÓN.....	18
1.2.22. NOMBRE DE OBJETOS DE BAJA	18
1.3. CONVENCIONES DE CÓDIGO	18
1.4. NORMAS REFERENTES AL USO DE MINÚSCULAS Y MAYÚSCULAS	29
1.4.1. ESTILOS DE GRAFÍA	29
1.4.2. REGLAS DE USO DE MAYÚSCULAS Y MINÚSCULAS PARA LOS IDENTIFICADORES	30
1.5. ANÁLISIS DE CÓDIGO	30
2. AUDITORIA	31
2.1. TABLAS AUDITABLES CON HISTÓRICO	31
2.2. TABLAS AUDITABLES SIN HISTÓRICO	32
2.3. TABLAS AUDITABLES POR SEGURIDAD.....	33
2.4. CAMPOS ADICIONALES PARA AUDITORÍA.....	33
2.5. CONSIDERACIONES DEL USO DE LOS CAMPOS DE AUDITORÍA	34
3. BIBLIOGRAFÍA	34
4. ANEXOS	34
4.1. ANEXO 1: PREFIJO DEL MÓDULO	34
4.2. ANEXO 2: CONFIGURACIONES DE EDITORES	35
4.3. ANEXO 3: MODELO DE DATOS DEL SIAT	36



SAT Servicio de Administración Tributaria de Lima	Tipo: DOCUMENTO TÉCNICO	Código: GIN-AGS-DT003 Versión: 01
	Título: ESTÁNDARES PARA EL DESARROLLO DE SISTEMAS EN SQL	Fecha de Vigencia: 16/05/2014
		Página: 4 de 37

1. ESTÁNDARES DE TRANSACT-SQL

1.1. NOMENCLATURA PARA LOS ELEMENTOS DE PROGRAMACIÓN

1.1.1. PREFIJOS DE TIPOS DE DATOS

Tipo de Dato	Tipo de SQL 2000 / 2008	Prefijo	Byte	Rango
Enteros	Bigint	bi	8 bytes	$2^{63} (-9.223.372.036.854.775.808)$ y $2^{63} - 1 (9.223.372.036.854.775.807)$.
	Int	i	4 bytes	$-2^{31} (-2.147.483.648)$ y $2^{31} - 1 (2.147.483.647)$
	Smallint	si	2 bytes	$-2^{15} (-32.768)$ y $2^{15} - 1 (32.767)$.
	Tinyint	ti	1 byte	0 y 255.
Lógico	Bit	b	Sí hay 8 ó menos columnas de tipo bit en una tabla, las columnas se almacenan como 1 byte. Si hay entre 9 y 16 columnas de tipo bit, se almacenan como 2 bytes y así sucesivamente.	1, 0 ó NULL.
Decimal y numeric	Decimal[(p[, s])]	de	5 bytes, 9 bytes, 13 bytes o 17 bytes	$-10^{38} +1$ y $10^{38} - 1$
	Numeric[(p[, s])]	n	5 bytes, 9 bytes, 13 bytes o 17 bytes	$-10^{38} +1$ y $10^{38} - 1$
Money y Smallmoney	Money	(sin uso)	8 bytes	$-263 (-922.337.203.685.477.5808)$ y $263 - 1 (+922.337.203.685.477.5807)$
	Smallmoney	(sin uso)	4 bytes	$-214.748.3648$ y $+214.748.3647$
Numéricos con aproximación	Float [(n)]	(sin uso)	4 bytes / 8 bytes	$-1.79E + 308$ a $-2.23E - 308$, 0 y de $2.23E - 308$ a $1.79E + 308$
	Real	r	4 bytes	$-3.40E + 38$ a $-1.18E - 38$, 0 y de $1.18E - 38$ a $3.40E + 38$
Fechas	Datetime	dt	8 bytes	Fecha y hora desde el 1 de enero de 1753 hasta el 31 de diciembre de 9999, con una precisión de 3,33 milisegundos o 0,00333 segundos
	Smalldatetime	sd	4 bytes	Fecha y hora desde el 1 de enero de 1900 hasta el 6 de junio de 2079
Cadenas de caracteres	Char[(n)]	c	n bytes	1 y 8.000 caracteres.
	Varchar[(n)]	v	n bytes	1 y 8.000 caracteres.
Cadenas de caracteres	Text	t	Cuando la página de códigos del servidor utiliza caracteres de doble byte, el almacenamiento sigue siendo de 2.147.483.647 bytes. Dependiendo de la cadena de caracteres, el espacio de almacenamiento puede ser inferior a 2.147.483.647 bytes.	Datos no Unicode de longitud variable, de la página de códigos del servidor y con una longitud máxima de 231-1 (2.147.483.647) caracteres.



 SAT <i>Servicio de Administración Tributaria de Lima</i>	Tipo: DOCUMENTO TÉCNICO	Código: GIN-AGS-DT003 Versión: 01
	Título: ESTÁNDARES PARA EL DESARROLLO DE SISTEMAS EN SQL	Fecha de Vigencia: 16/05/2014 Página: 5 de 37

Tipo de Dato	Tipo de SQL 2000 / 2008	Prefijo	Byte	Rango
Cadenas de caracteres Unicode	nChar(n)	nc	n* 2 bytes	Datos de carácter Unicode de longitud fija, con n caracteres. n debe estar comprendido entre 1 y 4.000
	nVarchar(n)	nv	n* 2 bytes	Datos de carácter Unicode de longitud variable, con n caracteres. n debe ser un valor comprendido entre 1 y 4.000.
	nText	nt	El tamaño del almacenamiento, en bytes, es dos veces el número de caracteres especificado	Datos Unicode de longitud variable con una longitud máxima de 230 - 1 (1.073.741.823) caracteres
Cadenas binarias	Binary [(n)]	bin	El tamaño de almacenamiento es n+4 bytes	Datos binarios de longitud fija de n bytes. El argumento n debe ser un valor comprendido entre 1 y 8.000
	Varbinary [(n)]	(sin uso)	El tamaño de almacenamiento es la longitud actual de los datos escritos + 4 bytes.	Datos binarios de longitud variable de n bytes. El argumento n debe ser un valor comprendido entre 1 y 8.000.
	Image	img	Desde 0 hasta 231-1 (2.147.483.647) bytes.	Datos binarios de longitud variable
Fechas	Time	tm	3 – 5	De 00:00:00.0000000 a 23:59:59.9999999
	Date	de	3	De 0001-01-01 a 9999-12-31
	Datetime2	dt2	6 – 8	De 0001-01-01 00:00:00.0000000 a 9999-12-31 23:59:59.9999999
	DateTimeOffset	dto	8 – 10	De 0001-01-01 00:00:00.0000000 a 9999-12-31 23:59:59.9999999 (en UTC)
XML	Xml	x		<i>La instancia de este tipo de dato no puede superar los 2 GB.</i>

Para otros tipos de objetos:

Tipo	Prefijo	Observaciones
Table	tab	Para las variables tipo Table (tabla)
Cursor	cur	Para las variables de tipo Cursor
Sql_variant	(sin uso)	Para los datos de tipo Variant
Timestamp	tms	Para los datos de tipo timestamp
Uniqueidentifier	unq	Para los datos de tipo uniqueidentifier.

Para la declaración de variables, el primer carácter del identificador debe ser uno de los siguientes símbolos:

Carácter	Uso
@	Representa a una variable local.
#	Representa el nombre de un objeto temporal. En el caso de una tabla o stored procedure que representa a un objeto temporal local.
##	Representan a un objeto temporal global.



 SAT <i>Servicio de Administración Tributaria de Lima</i>	Tipo: DOCUMENTO TÉCNICO	Código: GIN-AGS-DT003 Versión: 01
	Título: ESTÁNDARES PARA EL DESARROLLO DE SISTEMAS EN SQL	Fecha de Vigencia: 16/05/2014
		Página: 6 de 37

Cuando dos expresiones que disponen de tipos de datos diferentes, intercalaciones, precisión, escala o longitud los combina un operador:

- El tipo de datos de los valores resultantes viene determinado al aplicar las reglas de precedencia de tipos de datos a los tipos de datos de las expresiones de entrada.
- Si el tipo de datos del resultado es char, varchar, text, nchar, nvarchar o ntext, la intercalación del valor del resultado viene determinado por las reglas de precedencia de la intercalación.
- La precisión, escala y longitud del resultado dependen de la precisión, escala y longitud de las expresiones de entrada.

Para el caso de la intercalación, se deberá utilizar la 'Modern_Spanish_CI_AS' para todas las bases de datos implementadas y, por consiguiente, en las tablas de estas.

Por otro lado, el SQL proporciona sinónimos de tipos de datos para la compatibilidad con SQL-92. Sin embargo, estos sinónimos no deben emplearse.

Se recomienda que los nombres de los objetos temporales no excedan de los 20 caracteres, incluyendo al # o ##, debido a que SQL Server les agrega un sufijo numérico interno. Las tablas temporales deberán en lo posible tener el mismo nombre de la tabla que se extraen los datos y/o un nombre explícito sobre el proceso para la cual es necesitada. El prefijo para las tablas temporales deberá ser:

#tmp<Nombre de Tabla><Nombre Explícito>
#tmp<Nombre Explícito>

1.2. DECLARACIONES

Se debe evitar utilizar identificadores que entren en conflicto con las siguientes palabras clave:

Comando	Comando	Comando
Add	Except	Percent
All	Exec	Plan
Alter	Execute	Precision
And	Exists	Primary
Any	Exit	Print
As	Fetch	Proc
Asc	File	Procedure
Authorization	Fillfactor	Public
Backup	For	Raiserror
Begin	Foreign	Read
Between	Freetext	Readtext
Break	Freetexttable	Reconfigure
Browse	From	References
Bulk	Full	Replication
By	Function	Restore
Cascade	Goto	Restrict
Case	Grant	Return



SAT Servicio de Administración Tributaria de Lima	Tipo: DOCUMENTO TÉCNICO	Código: GIN-AGS-DT003 Versión: 01
	Título: ESTÁNDARES PARA EL DESARROLLO DE SISTEMAS EN SQL	Fecha de Vigencia: 16/05/2014
		Página: 7 de 37

Check	Group	Revoke
Checkpoint	Having	Right
Close	Holdlock	Rollback
Clustered	Identity	Rowcount
Coalesce	Identity_insert	Rowguidcol
Collate	Identitycol	Rule
Column	If	Save
Commit	In	Schema
Compute	Index	Select
Constraint	Inner	Session_user
Contains	Insert	Set
Containstable	Intersect	Setuser
Continue	Into	Shutdown
Convert	Is	Some
Create	Join	Statistics
Cross	Clave	System_user
Current	Kill	Table
Current_date	Left	Textsize
Current_time	Like	Then
Current_timestamp	Lineno	To
Current_user	Load	Top
Cursor	National	Tran
Database	Nocheck	Transaction
Dbcc	Nonclustered	Trigger.
Deallocate	Not	Truncate
Declare	Null	Tsequal
Default	Nullif	Union
Delete	Of	Unique
Deny	Off	Update
Desc	Offsets	Updatetext
Disk	On	Use
Distinct	Open	User
Distributed	Opendatasource	Values
Double	Openquery	Varying
Drop	Openrowset	View
Dummy	Openxml	Waitfor
Dump	Option	When
Else	Or	Where
End	Order	While
Errlvl	Outer	With
Escape	Over	Writetext
Xml	Time	Date



 SAT <i>Servicio de Administración Tributaria de Lima</i>	Tipo: DOCUMENTO TÉCNICO	Código: GIN-AGS-DT003 Versión: 01
	Título: ESTÁNDARES PARA EL DESARROLLO DE SISTEMAS EN SQL	Fecha de Vigencia: 16/05/2014
		Página: 8 de 37

1.2.1. NOMBRE DE BASES DE DATOS

<Siglas del Proyecto><Código de Distrito>

Para las siglas del proyecto, en los casos que este no sean siglas, se deberá seguir las reglas de uso de mayúsculas y minúsculas, deberá utilizar la convención Camel (ver Estilos de Grafia), de lo contrario, debe ser todo en mayúsculas. Además, no deben tener espacios en blanco ni el carácter underline ("_").

Por ejemplo:

Nombre de Base de datos	Descripción
SIAT001	Base de datos de No Tributario.
SIAT002	Base de datos de Tributario.

En los casos que la base de datos no contenga información de un distrito en particular, su nomenclatura será:

<BD>< Acrónimo de la Base de Datos>

Para este acrónimo, en los casos que este no sean siglas, se deberá seguir las reglas de uso de mayúsculas y minúsculas, deberá utilizar la convención Camel (ver Estilos de Grafia), de lo contrario, debe ser todo en mayúsculas. Además, no deben tener espacios en blanco.

Por ejemplo:

Nombre de Base de datos	Descripción
BDMTC	Base de datos con información del MTC
BCSUNARP	Base de datos con información de SUNARP

1.2.2. NOMBRE DE ESQUEMAS

<Nombre del Esquema>

Se deberá seguir las reglas de uso de mayúsculas y minúsculas, en este caso, utilizar la convención Camel (ver Estilos de Grafia). Además, no deben tener espacios en blanco, ni el carácter underline ("_") y ni caracteres especiales. Por ejemplo:



Nombre del Esquema	Descripción
Registro	Esquema para los objetos del sistema de Registro y Determinación de Deuda.
Reclamos	Esquemas para los objetos del sistema de Reclamos.

 SAT <i>Servicio de Administración Tributaria de Lima</i>	Tipo: DOCUMENTO TÉCNICO	Código: GIN-AGS-DT003 Versión: 01
	Título: ESTÁNDARES PARA EL DESARROLLO DE SISTEMAS EN SQL	Fecha de Vigencia: 16/05/2014
		Página: 9 de 37

1.2.3. NOMBRE DE TABLAS

<Prefijo de Módulo><Prefijo de Tipo de Tabla><Acrónimo de la Entidad>

Para las tablas de un modelo Dimensional, el nombre debe seguir el siguiente estándar:

<Prefijo de Tipo de Tabla><Nombre de la Entidad>

En donde, el “Nombre de la Entidad” puede contener el carácter “_” para separar palabras que componen o la describen.

Para el prefijo del módulo ver la tabla de módulos en el Anexo 1; para el caso de tipo de tablas, estas serán:

Diseño	Tipo	Prefijo	
Estándar	Maestro	Mae	
	Movimiento	Mov	
	Detalle	Det	
	Tabla	Tab	
	Imagen	Img	
Modelo Dimensional	Dimensión	Dim	D_
	Hecho	Hec	H_
	Look Up	Lkp	L_
	Transaccional	Trx	T_
	Stage		STG_

La siguiente tabla se lista algunas abreviaturas que se pueden usar dentro del **acrónimo de la Entidad**:

Abreviaturas	Descripción
<i>Deu</i>	<i>Deuda</i>
<i>Deu</i>	<i>Deuda</i>
<i>Doc</i>	<i>Documento</i>
<i>Dom</i>	<i>Domicilio</i>
<i>Est</i>	<i>Estado</i>
<i>Exp</i>	<i>Expediente</i>
<i>Lot</i>	<i>Lote</i>
<i>Pap</i>	<i>Papeleta</i>
<i>Per</i>	<i>Persona</i>
<i>Pre</i>	<i>Predio / Predial</i>
<i>Tip</i>	<i>Tipo</i>
<i>Tip</i>	<i>Tipo</i>
<i>Tmp</i>	<i>Temporal</i>
<i>Tra</i>	<i>Trámite</i>
<i>Veh</i>	<i>Vehículo / Vehicular</i>



Para el caso del Acrónimo de la Entidad deberá seguir las reglas de uso de

 SAT Servicio de Administración Tributaria de Lima	Tipo: DOCUMENTO TÉCNICO	Código: GIN-AGS-DT003 Versión: 01
	Título: ESTÁNDARES PARA EL DESARROLLO DE SISTEMAS EN SQL	Fecha de Vigencia: 16/05/2014
		Página: 10 de 37

mayúsculas y minúsculas, deberá utilizar la convención Pascal (ver Estilos de Grafía), por otro lado, se escribirá en plural siempre y cuando se trate de agrupar varios objetos de una entidad en particular. Además, no debe tener espacios en blanco. **Asimismo**, si el nombre de la Entidad está formado por una sola palabra, el acrónimo será esta palabra, por el contrario, si está formado por 2 palabras, se podrá usar la primera como prefijo, el cual puede ser de 3 caracteres. Si no cumple con los anteriores casos, se combinaran las siguientes longitudes, 3-3-+. Por ejemplo:

Tabla	Descripción
RDMaeDeuda	Maestro de Deuda
RDMovDeuda	Movimiento de Deuda
SGTabMotivos	Tabla de Motivos
IIDetIntercambio	Detalle del Intercambio
MTCMaeLicencia	Maestro de Licencias

Para **todos** los casos, las Tablas Históricas deberán finalizar con la letra "H" en mayúscula.

Por ejemplo:

Tabla	Descripción
RDMaeDeudaH	Histórico de Maestro de Deuda
PEMaePersonaH	Histórico de Maestro de Persona Extendido
MTCMaeLicenciaH	Histórico de Maestro de Licencias

Para los casos que se necesite crear una tabla extendida de otra, estas deberán finalizar con las letras "Ext".

Por ejemplo:

Tabla	Descripción
RDMaeDeudaExt	Maestro de Deuda Extendido
PEMaePersonaExt	Maestro de Persona Extendido
MTCMaeLicenciaExt	Maestro de Licencias Extendido

Por otro lado, solo para los casos de Tablas que están en un proceso de réplica como **tablas intermedias**, deberán de finalizar con las letras "_Rep", por ejemplo:

Tabla	Descripción
CCDomObl_Rep	Domicilio de Obligado (Replicado)
CCOb1Doc_Rep	Documentos del Obligado (Replicado)

Para el caso del nombre de la Entidad, este puede utilizar palabras completas, en un modelo dimensional. Por ejemplo:

Tabla	Descripción
DimDeuda	Deudas
DimMovimientoDeuda	Movimiento de Deuda



 SAT Servicio de Administración Tributaria de Lima	Tipo: DOCUMENTO TÉCNICO	Código: GIN-AGS-DT003 Versión: 01
	Título: ESTÁNDARES PARA EL DESARROLLO DE SISTEMAS EN SQL	Fecha de Vigencia: 16/05/2014
		Página: 11 de 37

<i>DimPersona</i>	<i>Personas</i>
<i>DimDomicilio</i>	<i>Domicilios</i>
<i>DimPapeleta</i>	<i>Papeletas</i>

Para todos los casos, el nombre de la tabla puede contener un máximo de 128 caracteres, excepto para los nombres de tablas temporales locales (nombres precedidos con un signo numérico simple #) que no pueden exceder de los 116 caracteres, para este caso, se recomienda que no se exceda de los 20 caracteres.

1.2.4. NOMBRE DE TABLAS TEMPORALES FÍSICAS

Tmp<Prefijo de Módulo><Acrónimo de la Información Almacenada>

Para el caso del Acrónimo de la Entidad deberá seguir las reglas de uso de mayúsculas y minúsculas, deberá utilizar la convención Pascal (ver Estilos de Grafito), por otro lado, se escribirá en plural siempre y cuando se trate de agrupar varios objetos de una entidad en particular. Además, no debe tener espacios en blanco. Por otro lado, si el nombre de la Entidad está formado por una sola palabra, el acrónimo será esta palabra, por el contrario, si está formado por 2 palabras, se podrá usar la primera como prefijo, el cual puede ser de 3 caracteres. Si no cumple con los anteriores casos, se combinaran las siguientes longitudes, 3-3-+. Por ejemplo:

Tabla	Descripción
<i>TmpCCRnkMenCoactiva</i>	<i>Temporal de Control y Cobranza del Ranking Mensual de Coactiva</i>
<i>TmpCCRnkSemCoaConductor</i>	<i>Temporal de Control y Cobranza Ranking Semanal de Coactiva del Conductor</i>

Estas tablas deben ser usadas por procesos masivos por lo cual almacenan información temporalmente.

Para este caso, el nombre de la tabla puede contener un máximo de 128 caracteres.

1.2.5. NOMBRE DE TABLAS TEMPORALES FÍSICAS PARA MIGRACIÓN

TmpMig<Prefijo de Módulo><Nombre Descriptivo del Objeto>

Ejemplo:

Tabla	Descripción
<i>TmpMigEJTablaMaestro</i>	<i>Temporal de Migración del módulo EJ para la tabla Maestro</i>
<i>TmpMigEJTablaDetalle</i>	<i>Temporal de Migración del módulo EJ para la tabla Detalle</i>

Estas tablas deben ser usadas por procesos masivos por lo cual almacenan información temporalmente.

Para este caso, el nombre de la tabla puede contener un máximo de 128 caracteres.



 SAT <i>Servicio de Administración Tributaria de Lima</i>	Tipo: DOCUMENTO TÉCNICO	Código: GIN-AGS-DT003 Versión: 01
	Título: ESTÁNDARES PARA EL DESARROLLO DE SISTEMAS EN SQL	Fecha de Vigencia: 16/05/2014 Página: 12 de 37

1.2.6. NOMBRE DE COLUMNAS

<Tipo de Dato SQL><Acrónimo del Atributo>

Para un campo que pertenezca a una tabla de un modelo Dimensional, el nombre debe seguir el siguiente estándar:

<Nombre del Atributo>

En donde, el “Nombre del Atributo” puede contener el carácter “_” para separar palabras que componen o la describen.

Para el caso del acrónimo del Atributo deberá seguir las reglas de uso de mayúsculas y minúsculas, que deberá utilizar la convención Camel (ver Estilos de Grafía). Por otro lado, si el campo está formado por una sola palabra, el acrónimo será esta palabra, por el contrario, si está formado por 2 palabras, se usara la primera como prefijo, que puede ser de 3 caracteres. Si no cumple con los anteriores casos, se combinaran las siguientes longitudes, 3-3-+. La longitud máxima del Identificador, en algún caso extremo, será de 15 caracteres. Además, no debe tener espacios en blanco ni el carácter underline (“_”) a excepción de los campos de una tabla dimensional. La siguiente tabla se lista algunas abreviaturas que se pueden usar dentro del nombre de una columna:

Abreviaturas	Descripción
Cod	Código
Deu	Deuda
Doc	Documento
Dom	Domicilio
Est	Estado
Exp	Expediente
Lot	Lote
Num	Número
Pap	Papeleta
Per	Persona
Tip	Tipo
Tra	Trámite
Tip	Tipo

Por ejemplo:

Campo	Descripción
siCodMunicipalidad	Código de Municipalidad
iCodPersona	Código de Persona
cNumDocDeuda	Número de Documento de Deuda
cNumPapeleta	Número de Papeleta

Para el caso del Nombre del Atributo, este puede ser palabras completas. Por ejemplo:



SAT Servicio de Administración Tributaria de Lima	Tipo: DOCUMENTO TÉCNICO	Código: GIN-AGS-DT003 Versión: 01
	Título: ESTÁNDARES PARA EL DESARROLLO DE SISTEMAS EN SQL	Fecha de Vigencia: 16/05/2014
		Página: 13 de 37

Campo	Descripción
<i>CodigoMunicipalidad</i>	<i>Código de Municipalidad</i>
<i>CodigoPersona</i>	<i>Código de Persona</i>
<i>NumeroDocumentoDeuda</i>	<i>Número de Documento de Deuda</i>
<i>NumeroPapeleta</i>	<i>Número de Papeleta</i>

1.2.7. NOMBRE DE CONSTRAINTS (PRIMARY KEY)

PK_<NOMBRE DE TABLA>

Todo el nombre deberá estar en mayúsculas. Además, no tener espacios en blanco.

Por ejemplo:

PK_RDMAEDEUDA
PK_RDMAEPERSONA

1.2.8. NOMBRE DE CONSTRAINTS (FOREIGN KEY)

FK_<NOMBRE DE TABLA ORIGEN>_<NOMBRE DE TABLA REFERENCIA>

Todo el nombre deberá estar en mayúsculas. Además, no debe tener espacios en blanco. La tabla de referencia es aquella cuyos campos deben existir en la tabla origen (existe una relación de referencia).

Por ejemplo:

FK_RDMAEDEUDA_SGMAEUSUARIOS
FK_RDMAEPERSONA_SGTIPDOCUMENTO

En los casos que existan más de una relación de la misma tabla en otra, se aumentará un correlativo al final del nombre comenzando del 1 con el formato "99".

Por ejemplo:

FK_RDMAEDEUDA_SGMAEUSU (Original o principal)
FK_RDMAEDEUDA_SGMAEUSU_01 (Secundario)
FK_RDMAEDEUDA_SGMAEUSU_02 (Secundario)

1.2.9. NOMBRE DE CONSTRAINTS (DEFAULT)

DF_<NOMBRE DE TABLA>_<NOMBRE DE CAMPO>

Todo el nombre deberá estar en mayúsculas. Además, no debe tener espacios en blanco.

Por ejemplo:

DF_RDMAEDEUDA_SDfecact
DF_RDMAEPERSONA_Cnomter



 SAT <i>Servicio de Administración Tributaria de Lima</i>	Tipo: DOCUMENTO TÉCNICO	Código: GIN-AGS-DT003 Versión: 01
	Título: ESTÁNDARES PARA EL DESARROLLO DE SISTEMAS EN SQL	Fecha de Vigencia: 16/05/2014

1.2.10. NOMBRE DE CONTRAINTES (CHECK)

Para un campo:

CKC_<NOMBRE DE TABLA>_<NOMBRE DE CAMPO>

Para una tabla

CKT_<NOMBRE DE TABLA> <NOMBRE DE CHECK>

Para los dos casos, todo el nombre deberá estar en mayúsculas. Además, no debe tener espacios en blanco.

1.2.11. NOMBRE DE ÍNDICES

IX_<NOMBRE DE TABLA>_<Correlativo por tabla>

Todo el nombre deberá estar en mayúsculas. Además, no debe tener espacios en blanco. El correlativo será de 3 caracteres y empezará en uno (1) rellenados con ceros a la izquierda.

Por ejemplo:

IX_RDMAEDEUDA_001
IX_RDMAEPERSONA_001

1.2.12. NOMBRE DE VISTAS

<Esquema>.<Prefijo de Módulo> VW <Acrónimo>

Para el caso del Acrónimo se deberá seguir las reglas de uso de mayúsculas y minúsculas, deberá utilizar la convención Pascal (ver Estilos de Grafía). Por otro lado, en el caso que la vista sea sobre una única tabla, se adopta el nombre de la tabla. Además, no debe tener espacios en blanco.

Este objeto deberá ser documentado con la siguiente estructura antes de su declaración:

```
-- ****  
-- <Descripción de la Vista>  
-- Input : <Parametros> - Descripcion de los parámetros  
-- Output : <Descripción de la Salida>  
-- Creado por : <Responsable>  
-- RQ : <Número de Requerimiento>  
-- Fec Creación : <Fecha Creación>  
-- Replicado : <Fecha Replicado> - <SERVIDOR>.<BASE DATOS>  
-----  
-- Fec Actualización : <Fecha de Actualización>  
-- Responsable : <Analista>  
-- RQ : <Número de Requerimiento>  
-- Motivo : <Motivo de la Modificación>  
-----
```



1.2.13. NOMBRE DE PROCEDIMIENTOS ALMACENADOS

<Esquema>,sp<Prefijo de Módulo>_<Nombre de Entidad Negocio> <Acción>

 SAT <i>Servicio de Administración Tributaria de Lima</i>	Tipo: DOCUMENTO TÉCNICO	Código: GIN-AGS-DT003 Versión: 01
	Título: ESTÁNDARES PARA EL DESARROLLO DE SISTEMAS EN SQL	Fecha de Vigencia: 16/05/2014
		Página: 15 de 37

Para el caso de la "Acción" se deberá seguir las reglas de uso de mayúsculas y minúsculas, deberá utilizar la convención Camel (ver Estilos de Grafix). Además, no debe tener espacios en blanco. La acción debe iniciar en un verbo infinitivo.

Por ejemplo:

```
dbo.spRD_Papeleta_Registrar
dbo.spTRD_DJPredial_Registrar
dbo.spRD_Persona_Consultar
dbo.spRD_Persona_BuscarPorTipo
```

Este objeto deberá ser documentado con la siguiente estructura:

```
-- ****
-- <Descripción del Procedimiento Almacenado>
-- Input      : <Parametros> - Descripción de los parámetros
-- Output     : <Descripción de la Salida>
-- Creado por : <Responsable>
-- RQ         : <Número de Requerimiento>
-- Fec Creación : <Fecha Creación>
-- Replicado  : <Fecha Replicado> - <SERVIDOR>.<BASE DATOS>
-----
-- Fec Actualización : <Fecha de Actualización>
-- Responsable       : <Analista>
-- RQ                 : <Número de Requerimiento>
-- Motivo             : <Motivo de la Modificación>
-- ****
```

1.2.14. NOMBRES DE FUNCIONES DEFINIDAS POR EL USUARIO

<Esquema>.fn<Prefijo de Módulo>_<Nombre de Entidad Negocio>_<Acción>

La Entidad se refiere a la Entidad de Negocio en donde se necesita realizar la Acción, el cual debe estar especificado en un diagrama de clases de una aplicación o solución. Para el caso de la Acción se deberá seguir las reglas de uso de mayúsculas y minúsculas, deberá utilizar la convención Camel (ver Estilos de Grafix). Además, no debe tener espacios en blanco. La acción debe iniciar con un verbo infinitivo.

Por ejemplo:

```
dbo.fnSG_Documento_Formatear
dbo.fnTRD_Deuda_DevolverDatos
dbo.fnCC_Papeleta_ValidarFechaEmision
```



Este objeto deberá ser documentado con la siguiente estructura antes de su declaración:

```
-- ****
-- <Descripción de la Función Definida por el Usuario >
-- Input      : <Parametros> - Descripción de los parámetros
-- Output     : <Descripción de la Salida>
-- Creado por : <Responsable>
-- RQ         : <Número de Requerimiento>
-- Fec Creación : <Fecha Creación>
-- Replicado  : <Fecha Replicado> - <SERVIDOR>.<BASE DATOS>
-----
-- Fec Actualización : <Fecha de Actualización>
-- Responsable       : <Analista>
-- RQ                 : <Número de Requerimiento>
-- Motivo             : <Motivo de la Modificación>
-- ****
```

 SAT <i>Servicio de Administración Tributaria de Lima</i>	Tipo: DOCUMENTO TÉCNICO	Código: GIN-AGS-DT003 Versión: 01
	Título: ESTÁNDARES PARA EL DESARROLLO DE SISTEMAS EN SQL	Fecha de Vigencia: 16/05/2014
		Página: 16 de 37

1.2.15. NOMBRES DE TRIGGERS

Para Inserción:

<Esquema>.ti<Prefijo de Módulo>_<Nombre de tabla>

Para Actualización:

<Esquema>.tu<Prefijo de Módulo>_<Nombre de tabla>

Para Eliminación:

<Esquema>.td<Prefijo de Módulo>_<Nombre de tabla>

Para Auditoría:

<Esquema>.ta<Prefijo de Módulo>_<Nombre de tabla>

El nombre de la tabla debe seguir el estándar de este. Por otro lado, no debe tener espacios en blanco.

Este objeto deberá ser documentado con la siguiente estructura antes de su declaración:

```
-- ****
-- <Descripción del Trigger>
-- Creado por      : <Responsable>
-- RQ              : <Número de Requerimiento>
-- Fec Creación    : <Fecha Creacion>
-- Replicado       : <Fecha Replicado> - <SERVIDOR>.<BASE DATOS>
-----
-- Fec Actualización : <Fecha de Actualizacion>
-- Responsable      : <Analista>
-- RQ                : <Número de Requerimiento>
-- Motivo            : <Motivo de la Modificación>
-- ****
```

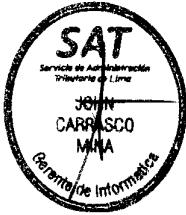
1.2.16. NOMBRES DE LOS PARÁMETROS

@p<Tipo de Dato SQL><Nombre de Campo / Acrónimo>

Para el caso del nombre del campo debe seguir las reglas correspondientes a este. En el caso del Acrónimo deberá seguir las reglas de uso de mayúsculas y minúsculas, deberá utilizar la convención Camel (ver Estilos de Grafito). Por otro lado, si el campo está formado por una sola palabra, el acrónimo será la palabra, en otro caso, si está formado por 2 palabras, se usara el primer prefijo, que puede ser de 3 caracteres; y si está formado por más de dos palabras, se combinaran las siguientes longitudes, 3-3-+. La longitud máxima del Identificador, en algún caso extremo, será de 15 caracteres como máximo. Además, no debe tener espacios en blanco.

Por ejemplo:

Campo	Descripción
@psiCodMunicipalidad	Código de Municipalidad
@piCodPersona	Código de Persona
@pcNumDocDeuda	Número de documento de deuda
@pcNumPapeleta	Papeleta



 SAT <i>Servicio de Administración Tributaria de Lima</i>	Tipo: DOCUMENTO TÉCNICO	Código: GIN-AGS-DT003 Versión: 01
	Título: ESTÁNDARES PARA EL DESARROLLO DE SISTEMAS EN SQL	Fecha de Vigencia: 16/05/2014 Página: 17 de 37

1.2.17. NOMBRES DE VARIABLES

@<Tipo de Dato SQL><Acrónimo>

Si la variable se refiere a un campo de una tabla esta deberá tomar el nombre del campo. En el caso del Acrónimo deberá seguir las reglas de uso de mayúsculas y minúsculas, deberá utilizar la convención Camel (ver Estilos de Grafía). Por otro lado, si el campo está formado por una sola palabra, el acrónimo será la palabra, en otro caso, si está formado por 2 palabras, se usara el primer prefijo, que puede ser de 3 caracteres; y si está formado por más de dos palabras, se combinaran las siguientes longitudes, 3-3-+. La longitud máxima del Identificador, en algún caso extremo, será de 15 caracteres como máximo. Además, no debe tener espacios en blanco. Por ejemplo:

Campo	Descripción
@siCodMunicipalidad	Código de Municipalidad
@iCodPersona	Código de Persona
@cNumDocDeuda	Número de documento de deuda
@cNumPapeleta	Número de Papeleta
@iCorrelativo	Correlativo

Para el caso de variables que son usados como contador dentro de bucle (WHILE) estos pueden utilizar como nombre una letra del abecedario a partir de la letra "i". Ejemplo: @i, @j, @k, etc.

1.2.18. NOMBRES DE ARCHIVOS XML

<Módulo><Descripción>

Los archivos xml utilizados deben tener como parte inicial el prefijo del módulo al que pertenece y finalmente una descripción o el nombre de la entidad a la que pertenece, para esto utilizar la convención Pascal. Por ejemplo: RDMenu.xml.

1.2.19. NOMBRES DE PAQUETES

DTSX_<Módulo>_<Nombre del paquete>

El nombre del paquete debe describir la acción o proceso que realiza. Asimismo, se deberá utilizar la convención Pascal y el nombre no deberá tener espacios en blanco ni caracteres extraños. Por ejemplo: DTSX_BI_CargarDatosDJPredial.



1.2.20. NOMBRES DE TAREAS (JOB)

JOB_<Módulo>_<Nombre del paquete>

El nombre de las tareas debe describir la acción o proceso que realiza. Por otro lado, se deberá utilizar la convención Pascal y el nombre no deberá tener espacios en blanco ni caracteres extraños. Por ejemplo: JOB_CC_GenerarCopiaPIT_NoTributaria_Lote_9888

 SAT <i>Servicio de Administración Tributaria de Lima</i>	Tipo: DOCUMENTO TÉCNICO	Código: GIN-AGS-DT003 Versión: 01
	Título: ESTÁNDARES PARA EL DESARROLLO DE SISTEMAS EN SQL	Fecha de Vigencia: 16/05/2014
		Página: 18 de 37

1.2.21. NOMBRES DE INICIO DE SESIÓN

Usr<Nombre descriptivo del usuario>

El nombre de inicio de sesión de SQL, es decir, el usuario SQL para acceso al servidor, debe describir lo que realizará dentro de la base de datos. Por otro lado, se deberá utilizar la convención Pascal y el nombre no deberá tener espacios en blanco ni caracteres extraños.

1.2.22. NOMBRE DE OBJETOS DE BAJA

Tmp<Prefijo de Módulo>_<Nombre del Objeto Original>

Por ejemplo:

Tabla	Descripción
<i>TmpGD_GeDatVTr</i>	<i>Temporal de baja de la tabla “GeDatVTr”</i>
<i>TmpRD_RDTabTEF</i>	<i>Temporal de baja de la tabla “RDTabTEF”</i>
<i>TmpGD_spGe_TramiteRestaurarDeuda</i>	<i>Temporal de baja del SP “spGe_TramiteRestaurarDeuda”</i>
<i>TmpGD_spTGD_Actualiza_PagoCompensacion</i>	<i>Temporal de baja del SP “spTGD_Actualiza_PagoCompensacion”</i>

Para el caso de tablas dados de baja, no es necesario renombrar los objetos relacionados directamente a estos (PK, DF, CHK, IX ni FK).

1.3. CONVENCIONES DE CÓDIGO

Estas convenciones tienen los propósitos de fácil lectura y comprensión del código, así como facilitar su mantenimiento, por lo que se debe respetar al programar.

Nomenclatura:

- Ninguna variable u objeto de la base de datos llevará como parte del nombre caracteres especiales (#, \$ %, &, etc.), así como tampoco tildes (').
- En todas las palabras reservadas se usará la primera letra en mayúsculas y el resto en minúsculas. Estas se hacen extensivas a las funciones propias del SQL.
- Los nombres de los objetos de base de datos y variables deben estar en idioma español.

Diseño:

- Sangría de 3 caracteres (espacios), las tabulaciones en los editores de código SQL deben configurarse para que graben las tabulaciones en espacios.
- Usar una línea de código para separar grupos lógicos de código.
- Utilizar una sola instrucción por línea.
- Utilizar solo una declaración por línea.
- Usar la siguiente estructura para los procedimientos almacenados, funciones de usuario, vistas y triggers:



 SAT <i>Servicio de Administración Tributaria de Lima</i>	Tipo: DOCUMENTO TÉCNICO	Código: GIN-AGS-DT003 Versión: 01
	Título: ESTÁNDARES PARA EL DESARROLLO DE SISTEMAS EN SQL	Fecha de Vigencia: 16/05/2014
		Página: 19 de 37

Código	It.
USE <Base de Datos> GO	1
SET QUOTED_IDENTIFIER ON GO SET ANSI_NULLS ON GO	2
-- ***** -- <Descripción de la que realiza el objeto> -- Input : <Parametros> - Descripción de los parámetros -- Output : <Descripción de la Salida> -- Creado por : <Responsable> -- RQ : <Número de Requerimiento> -- Motivo : <Motivo de la creación> -- Fec Creación : <Fecha Creación> -- Replicado : <Fecha Replicado> - <SERVIDOR>.<BASE DATOS> ----- -- Fec Actualización : <Fecha de Actualización 1> -- Responsable : <Analista 1> -- RQ : <Número de Requerimiento> -- Motivo : <Motivo de la Modificación 1> ----- -- Fec Actualización : <Fecha de Actualización 2> -- Responsable : <Analista 2> -- RQ : <Número de Requerimiento> -- Motivo : <Motivo de la Modificación 2> ----- /* <Código ejemplo> */	3
[Create / Alter] [Procedure / Function / Trigger / View] <Esquema>.<Nombre>	4
(<Parámetros>)	5
As Begin Set Nocount On <Cuerpo del objeto> Set Nocount Off End	6
GO SET QUOTED_IDENTIFIER OFF GO SET ANSI_NULLS OFF GO	7
	8

En donde:



- 1- Especificar la Base de datos de donde pertenecerá o pertenece el objeto **en mayúsculas**.
- 2- Cláusulas o sentencias de inicio del objeto **en mayúsculas**.
- 3- Documentación del objeto; en el caso del campo de replicado se llenará solo si el objeto está replicado, en caso contrario se dejará vacío.
- 4- Código de ejemplo (solo para los casos de SP y funciones).
- 5- Creación o modificación del objeto.
- 6- Declaración de Parámetros.
- 7- Cuerpo del objeto, siempre debe iniciar con un "Begin" con su respectivo "Set Nocount On" y terminar con un "Set Nocount Off" y su "End". **Esto puede cambiar para algunos objetos de la base de datos.**

 SAT <i>Servicio de Administración Tributaria de Lima</i>	Tipo: DOCUMENTO TÉCNICO	Código: GIN-AGS-DT003 Versión: 01
	Título: ESTÁNDARES PARA EL DESARROLLO DE SISTEMAS EN SQL	Fecha de Vigencia: 16/05/2014
		Página: 20 de 37

8- Cláusulas o sentencias de fin del objeto *en mayúsculas*.

Para el caso del “Analista” se debe especificar los nombres y apellidos del personal que trabaja en la institución, en el caso de que el objeto es creado o modificado por un tercero, debe aparecer el nombre completo de este.

Comentarios:

- No utilizar comentarios al final de una instrucción con excepción a las declaraciones.
- Colocar el comentario en la línea superior del código que corresponda el comentario.
- Comenzar el texto del comentario con una letra mayúscula.
- Finalizar el comentario con un punto.
- Inserta un espacio entre el delimitador de comentario /*, --) y el comentario.
- **Los comentarios deben estar en idioma español.**
- Los comentarios deben escribirse con una gramática y puntuación adecuada. Se debe escribir de manera clara y concisa para transmitir la mayor cantidad de información posible.
- Se deben emplear comentarios en los siguientes casos:
 - Si se declara variable, procedimiento almacenado, función definida por el usuario, etc., sin ponerle un nombre explícito.
 - Si se usa una lógica compleja.
 - Si se inicializa una variable en un valor que no sea 0, -1, etc.
 - En casos que el programador determine adecuado.

Lenguaje:

- Palabras reservadas:
 - o Las palabras reservadas Begin - Else - End deberán estar en la misma columna de indentación de la sentencia lógica que las origina. Cuando los identificadores correspondientes Begin – End se extiendan demasiado en el código deberá comentarse en el identificador End la línea de la cual proviene, esto hará más fácil el seguimiento de la codificación.
 - o Cada sentencia debe ir en una línea diferente.
- Case:
 - o Los Case deben estar alineados de la siguiente manera: el Case en una línea y los When y el Else cada uno en otra línea diferente.

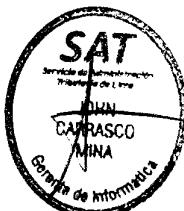
Por ejemplo:

```
Case
  When .... Then ....
  Else .....
End
```

- o Si el Then del Case fuera demasiado largo se indentará a 3 espacios adicionales al When al que pertenece:

Por ejemplo:

```
Case
  When ....
```



 SAT <i>Servicio de Administración Tributaria de Lima</i>	Tipo: DOCUMENTO TÉCNICO	Código: GIN-AGS-DT003 Versión: 01
	Título: ESTÁNDARES PARA EL DESARROLLO DE SISTEMAS EN SQL	Fecha de Vigencia: 16/05/2014
	Página: 21 de 37	

```

    Then .....
    Else .....
End
  
```

- Join:

- o Los grupos de claves para los Join también deben estar indentados si el ancho supera las 140 columnas.

- Where:

- o El Where cada condicionante debe ir en una línea aparte y el And deberá estar alineado a la derecha de la sentencia Where

Por ejemplo:

```

Where siCodMun = @psiCodMun
  And cNumdoc = @psiCodMun
  And siTipEdo = 1
  And siCodUsu = 457
  
```

- From:

- o Las sentencias From al incluir varias tablas deberán ser complementadas con las sentencias Inner Join, Left Join o Right Join según sea el caso. Considerar que el ancho de las columnas para la codificación no debe exceder de 140.
- o No se debe utilizar el siguiente código:

```

From RDMAeDeu a, RDMovDeu b, SGMAeUsu c
Where a.sicodmun = b.sicodmun
  And a.cnumdoc = b.cnumdoc
  And a.sicodusu *= c.sicodusu
  
```

A cambio se deberá usar el Join, por ejemplo:

```

From dbo.RDMAeDeu a
Inner Join dbo.RDMovDeu b On (a.siCodMun = b.siCodMun And
a.cNumDoc = b.cNumDoc)
Left Join dbo.SGMaeUsu c On (a.siCodUsu = c.siCodUsu)
  
```

- Sugerencia de tabla

- o Cuando la sugerencia se especifica con otra opción, debe especificarse con la palabra clave WITH.

Por ejemplo:

```

From dbo.RDMAeDeu a
Inner Join dbo.RDMovDeu b With (Nolock, Index=PK_RDMOVDEU)
On (a.siCodMun = b.siCodMun And a.cNumDoc = b.cNumDoc)
  
```

- "DISTINCT" y "ORDER BY"

- o Cuando se utilice estas dos sentencias en un SELECT se debe tener mucho cuidado en el alias de los campos que están en el ORDER BY ya que estos deben ser parte de la consulta.



SAT <i>Servicio de Administración Tributaria de Lima</i>	Tipo: DOCUMENTO TÉCNICO	Código: GIN-AGS-DT003 Versión: 01
	Título: ESTÁNDARES PARA EL DESARROLLO DE SISTEMAS EN SQL	Fecha de Vigencia: 16/05/2014
	Página: 22 de 37	

Por ejemplo:

```

Select Distinct
  cOrigen = XX.cOrigen,
  DctoNro = RTrim(LTrim(XX.DctoNro)),
  sdFecDoc = Convert(Char(10),XX.sdFecDoc,103)
From #tmpDocRelTramite XX
Where XX.DctoNro Is Not Null
Order By XX.cNumTra, XX.sdFecDoc Desc
  
```

La sentencia anterior tiene error en el alias del campo "sdFecDoc" ya que este no pertenece a la tabla temporal "#tmpDocRelTramite". El siguiente código corrige el problema anterior:

```

Select Distinct
  cOrigen = XX.cOrigen,
  DctoNro = RTrim(LTrim(XX.DctoNro)),
  sdFecDoc = Convert(Char(10),XX.sdFecDoc,103)
From #tmpDocRelTramite XX
Where XX.DctoNro Is Not Null
Order By XX.cNumTra, sdFecDoc Desc
  
```

Adicionalmente, se debe tener en cuenta que al usar el ORDER BY se debe especificar el nombre del campo.

- Esquemas:

- En todas las sentencias T-SQL, los objetos de las base de datos deben ir con el esquema de origen. Si la base de datos no trabaja con un esquema, como es el caso del SQL 2000, se debe poner el esquema por defecto "dbo".

- Convertir datos:

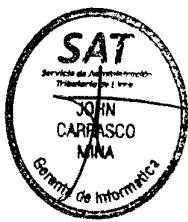
- Cuando se quiera convertir un dato de Varchar o Char a Datetime o Smalldatetime evitar el uso de la función Cast, en su lugar se utilizará la función Convert con el formato 103.

```

Select Cast('28/04/2009' As Smalldatetime)
Select Convert(Smalldatetime, '28/04/2009', 103)
  
```

- Bloqueos:

- Se recomienda no tener procesos transaccionales que demoren demasiado, para evitar así bloqueos en otros procesos.
- Se recomienda, en procesos transaccionales, el bloqueo por registro, para esto se debe utilizar la cláusula "ROWLOCK".
- Las sugerencias de no bloqueo que se recomienda en la lectura (Select) es la sentencia Nolock. Esta "no emite bloqueos compartidos y no respeta los bloqueos exclusivos" por lo cual "es posible leer una transacción no confirmada o un conjunto de páginas deshechas en mitad de una lectura" (lecturas fantasma). Es por esto que su uso debe ser en tablas de bajo nivel transaccional.



SAT <small>Servicio de Administración Tributaria de Lima</small>	Tipo: DOCUMENTO TÉCNICO	Código: GIN-AGS-DT003 Versión: 01
	Título: ESTÁNDARES PARA EL DESARROLLO DE SISTEMAS EN SQL	Fecha de Vigencia: 16/05/2014
		Página: 23 de 37

Por ejemplo:

```
From dbo.RDMaeDeu a
Inner Join dbo.RDMovDeu b On (a.siCodMun = b.siCodMun And
a.cNumDoc = b.cNumDoc)
Left Join dbo.SGMaeUsu (Nolock) c On (a.siCodUsu =
c.siCodUsu)
```

- **Eliminación de objetos:**

- **Cuando sea necesario eliminar objetos de la base de datos, se debe usar la función “Object_Id”. Por ejemplo:**

```
USE BaseDatos;
GO
If Object_Id(N'dbo.Tabla', N'U') Is Not Null
Begin
    Drop Table dbo.Tabla;
End
```

En el caso de que no se pueda usar la sentencia “USE”:

```
If Object_Id(N'BaseDatos.dbo.Tabla', N'U') Is Not Null
Begin
    Drop Table dbo.Tabla;
End
```

Directrices:

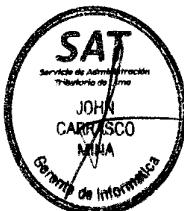
- General:

- Uso de la sentencia USE, esta sentencia se debe incluir antes de la cabecera en todos los procedimientos almacenados, funciones de usuario, vistas, triggers, scripts, etc. que se vayan a consultar, crear y/o actualizar datos el cual acompañara la base de datos en donde realizará el proceso.
- Se recomienda el uso de una cadena con datos XML como parámetro o parámetros cuando el SP necesite datos estructurados o de múltiples parámetros y cuando el proceso sea complejo, de lo contrario es más óptimo utilizar los tipos de parámetros comunes.
- **No se deben usar objetos de base de datos encriptados, esta regla se puede obviar por motivos de seguridad de información que estén debidamente sustentada.**
- **No usar sentencias “DROP” a objetos de la base de datos, solo será permitido esta sentencia cuando se realice cambios al modelo de datos mediante un script T-SQL.**
- **No usar sentencias “GRANT” a objetos de la base de datos, solo será permitido esta sentencia cuando se permisos de un usuario SQL por medio de un script T-SQL.**
- **No utilizar la sentencia “OPENROWSET” en objetos de la base de datos.**
- **No utilizar la sentencia “BULK INSERT” en objetos de la base de datos.**



SAT <small>Servicio de Administración Tributaria de Lima</small>	Tipo: DOCUMENTO TÉCNICO	Código: GIN-AGS-DT003 Versión: 01
	Título: ESTÁNDARES PARA EL DESARROLLO DE SISTEMAS EN SQL	Fecha de Vigencia: <u>16/05/2014</u> Página: 24 de 37

- ***En la creación de una tabla se debe incluir dentro de la sentencia "CREATE TABLE" la creación de los DEFAULT y CHECK de los campos y el PK de la tabla.***
- Archivos:
 - ***Cualquier archivo que contenga sentencias T-SQL los cuales serán ejecutadas en un servidor SQL Server debe tener la extensión "sql".***
 - Tratar de evitar archivos muy grandes, si un archivo tiene más de 500~700 líneas de código, se podría reconsiderar recrearlo en diferentes procedimientos almacenados o funciones utilitarias.
- Consultas:
 - En la consulta de datos de una tabla o grupo de ellas (físicas o temporales) se debe especificar los campos que intervienen en dicha consulta.
 - ***En la consulta se debe especificar el alias de la tabla al que pertenece un campo.***
- ***Insertar datos:***
 - ***Al llenar por primera vez una tabla que tengan un campo de tipo identidad, a este campo no debe faltarle ningún correlativo, es decir, el valor del campo identidad del último registro debe coincidir con la cantidad de registros insertados.***
 - ***No utilizar la opción "SET IDENTITY_INSERT".***
- Actualización de datos:
 - Para la actualización de datos, por cualquier proceso, se debe actualizar los campos de auditoría.
 - Para la eliminación de datos, ***no se debe hacer uso de la sentencia "DELETE" para ello*** se deberá usar un campo que indique la activación o no del registro (bActivo de tipo Bit el cual debe tener por defecto el valor '1'). En los casos que no se pueda seguir esta regla, se deberá almacenar los datos en una tabla histórica, con la misma estructura que la tabla de donde proviene los datos (revisar el punto 2.1 - Tablas Auditables con Histórico).
 - En la inserción de datos a una tabla (física o temporal) se debe especificar los campos que intervienen en esta inserción.
- ***Migración de datos:***
 - ***En las tablas que tengan registros migrados, se debe activar el campo "Migrado?" ("bMigrado" de tipo Bit el cual debe tener por defecto el valor '0').***



 SAT <i>Servicio de Administración Tributaria de Lima</i>	Tipo: DOCUMENTO TÉCNICO	Código: GIN-AGS-DT003 Versión: 01
	Título: ESTÁNDARES PARA EL DESARROLLO DE SISTEMAS EN SQL	Fecha de Vigencia: 16/05/2014
		Página: 25 de 37

Prácticas:

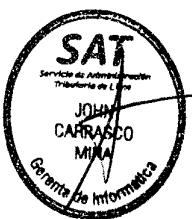
- Diseño:
 - Un modelo de datos *para un sistema transaccional* deberá cumplir, como mínimo, la tercera forma normal (3FN). *Esta regla se puede omitir por motivos de optimización, rendimiento debidamente fundamentados o para modelos Dimensionales.*
 - La integridad referencial de los datos deberá hacerla la base de datos con el uso de constraint y de FK definidos. Esta regla se puede omitir por motivos de optimización o rendimiento debidamente fundamentados.
 - *Todos los constraint de una tabla deben estar “habilitados” (“check”). Esta regla se puede omitir por motivos de optimización o rendimiento debidamente fundamentados.*
 - *No se debe hacer uso de la sentencia “with nocheck”. Esta regla se puede omitir por motivos de optimización o rendimiento debidamente fundamentados.*
 - *Para el caso de tablas de solo lectura y que no pertenezcan al negocio del sistema estas pueden ser almacenadas en otra base de datos.*
 - *Los campos de tipo descripción en una tabla tipo o estados por ejemplo, no deberán permitir nulos.*
 - Cada FK debe tener su correspondiente índice, esta regla se puede omitir por motivos de optimización o rendimiento.
 - Los campos que hacen referencia a otras tablas deberán tener el nombre de la tabla origen, esto puede obviarse cuando se repita más de una vez dicho campo, *pero el nombre original debe ser parte del nuevo nombre. Ejemplo:*
 - siCodUsu – Código de Usuario*
 - siCodUsuCreación – Código Usuario de Creación*
 - Evitar en lo posible agregar campos a tablas críticas de un sistema *en mantenimiento* puesto que el riesgo de una modificación de este tipo es alta. Por lo tanto, se debe “extender” el modelo; para esto, se deberá crear una tabla con relación de 1 a 1 con la principal. *Asimismo, el nombre de la tabla extendida debe iniciar con el mismo nombre de la tabla principal y adicionar el sufijo “Ext”, cabe indicar que la relación de estas dos tablas debe ser de uno a uno.*
 - Los tipos de datos Varchar se utilizará solo para longitudes mayores a 20 caracteres, normalmente aplicadas a campos de descripción, nombres, etc., en el cual no se sabe el tamaño que ocupa un texto determinado.
 - *Dentro de un criterio o filtro “WHERE”, cualquier campo debe estar filtrado por una variable del mismo tipo, la cual debe contener el valor a filtrar. En el caso de tener valores a convertir, estos deben realizarse antes de aplicar el criterio.*
 - Tener en cuenta las siguientes recomendaciones al crear campos de tipo



 SAT <i>Servicio de Administración Tributaria de Lima</i>	Tipo: DOCUMENTO TÉCNICO	Código: GIN-AGS-DT003 Versión: 01
	Título: ESTÁNDARES PARA EL DESARROLLO DE SISTEMAS EN SQL	Fecha de Vigencia: 16/05/2014
		Página: 26 de 37

binarios:

- “Una columna a la que se asigne el tipo de datos binary debe tener la misma longitud fija (hasta 8 KB) para cada fila. En una columna a la que se asigne el tipo de datos varbinary, las entradas pueden variar en el número de dígitos hexadecimales (hasta 8 KB) que contienen.”
- “Las columnas con datos image pueden utilizarse para almacenar datos binarios de longitud variable que excedan los 8 KB, como documentos de Microsoft Word®, hojas de cálculo de Microsoft Excel® e imágenes que incluyan archivos de mapa de bits, archivos con formato GIF (Graphics Interchange Format) y archivos con formato JPEG (Joint Photographic Experts Group).”
- “En general, utilice varbinary para almacenar datos binarios, excepto si su longitud supera los 8 KB, en cuyo caso deberá utilizar el tipo de datos image. Es recomendable que la longitud definida de una columna binaria no supere la longitud máxima prevista para los datos binarios que deben almacenarse.”
- En los casos que una tabla tenga un campo identidad, este debe ser su clave primaria (PK), **no se permite claves compuestas que contenga un campo identidad**.
- **Un campo identidad debe iniciar siempre en 1 y con un incremento de 1.**
- **No usar un campo identidad para una tabla del sistema.**
- **No se debe usar los valores por defecto generales en la base de datos “CREATE DEFAULT”, En su lugar, usar definiciones predeterminadas creadas con la palabra clave DEFAULT de ALTER TABLE o CREATE TABLE.**
- Antes de crear un índice a una tabla se debe verificar lo siguiente:
 - Si existe uno que contemple una parte del índice a crear, se deberá utilizar o modificarlo para su nuevo uso. Por ejemplo: si se necesita crear un índice por los campos Campo1, Campo2 y Campo3, pero en la tabla ya existe un índice por Campo1 y Campo2, se deberá modificar este adicionándole el Campo3.
 - Si existe un índice que tengan los mismos campos, pero en distinto orden, se deberá utilizar este para los propósitos requeridos.
- **Cualquier inserción o actualización masiva de datos a una tabla o más tablas, se debe realizar mediante un script T-SQL. Solo en los casos de migración de datos, se podrá utilizar un paquete de SQL Server (ETL), siguiendo los criterios para ello.**
- **Para poder utilizar un paquete de SQL Server (ETL) para migración de datos, se deberá tener en cuenta los siguientes criterios:**
 - **Que los datos a procesar superen el millón de registros o que por motivos técnicos no se pueda cargar la data utilizando un script T-SQL.**



SAT <small>Servicio de Administración Tributaria de Lima</small>	Tipo: DOCUMENTO TÉCNICO	Código: GIN-AGS-DT003 Versión: 01
	Título: ESTÁNDARES PARA EL DESARROLLO DE SISTEMAS EN SQL	Fecha de Vigencia: 16/05/2014
		Página: 27 de 37

- *Que los datos provengan de fuentes externas o fuentes no controlados por el área responsable.*
- *Que el resultado final de los datos migrados sean almacenados en una o varias Tablas Temporales Físicas para Migración.*
- *Al crear una Tabla Temporal Física para Migración se deberá tener en cuenta lo siguiente:*
 - *No será necesario normalizar la tabla.*
 - *Debe estar crear en la base de datos que almacenará los datos finales.*
 - *No se podrá actualizar la data ("update") durante el proceso de migración.*
 - *No se podrá eliminar la data ("delete" / "truncate") durante el proceso de migración.*
 - *No se podrá modificar ("alter") ni eliminar la tabla ("drop table") durante el proceso de migración.*

- Código:

- Para evitar "código en duro", hacer uso de tablas de parámetros del sistema o del módulo, a esto se incluye los valores o mensajes que se muestran en la interfaz de usuario (GUI).
- Los filtros a campos de tablas deberán utilizar el mismo tipo de dato que el campo filtrado.
- No se debe utilizar la consultas totales usando '*' en las sentencias Select. En los casos de utilizar la cláusula Exists de una consulta, esta debe tener solo un campo (de preferencias el campo clave de la tabla).
- Eliminar todas las tablas temporales utilizadas en un proceso (tarea, jobs, DTS, procedimientos almacenados, etc.).
- ***No utilizar tablas temporales globales (##) en los procedimientos almacenados ni en los desencadenadores.***
- Utilizar tablas temporales (#) para los casos de consultas muy grandes y solo el uso de variables de tipo Table para consultas que devuelvan pocos datos.
- ***Si una tabla temporal necesita un índice, esta se debe crearse luego de haber llenado la tabla temporal.***
- No usar Cursos, para esto evaluar el uso de tablas temporales o variables de tipo Table, dependiendo de la cantidad de filas a trabajar.
- Los alias de campos deben ser declarados con la instrucción "As".
- No se deben incluir sentencias comentadas, si no se usan deben eliminarse.



 SAT <i>Servicio de Administración Tributaria de Lima</i>	Tipo: DOCUMENTO TÉCNICO	Código: GIN-AGS-DT003 Versión: 01
	Título: ESTÁNDARES PARA EL DESARROLLO DE SISTEMAS EN SQL	Fecha de Vigencia: 16/05/2014
		Página: 28 de 37

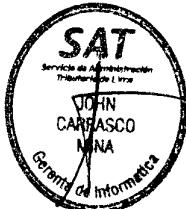
- La asignación de variables se deben hacer utilizando la sentencia "Set", salvo en la asignación con valores de tablas con "Select".
- En los procedimientos almacenados **y desencadenadores (TRIGGER)** se debe incluir la sentencia "Set Nocount On" al inicio y la sentencia "Set Nocount Off" al final, por cuestiones de performance.
- Todas las sentencias Where debe respetar la estructura de índices de la tabla al igual que los Join.
- **El uso de la sentencia "Set ForcePlan On" debe ser por motivos de performance.**
- **En las sentencias JOIN no utilizar variables o parámetros, para ello se debe usar los campos de cada tabla. Solo en los casos de performance se puede omitir esta regla.**
- En un procedimiento almacenado no se deberá iniciar, deshacer ni confirmar una transacción (Begin Transaction / Rollback Transaction / Commit Transaction). Esta regla no se aplicará en procesos que por problemas de performance necesiten manejar una transacción.
- En el caso de los procedimientos almacenados que son exclusivamente para mostrar información se deberá evitar lógicas como la siguiente:

```

Create Procedure dbo.spSG_General_Consultar
(
    @psiTipoCon Smallint,
    @pcCodigo Char(15)
)
As
Begin
    Set Nocount On

    If @psiTipoCon = 1
        Select siCampo1, cCampo2
        From dbo.Tabla
    Else
        Select siCampo1, cCampo2
        From dbo.Tabla
        Where cCodigo = @pcCodigo

    Set Nocount Off
End
  
```



No es óptimo crear procedimientos almacenados con este tipo de dato porque el SQL guarda una estadística de ejecución cada vez que se use este procedimiento, es decir, que si la primera vez se le pasó el parámetro '@psiTipoCon = 1', el SQL no empleará índices para el Select; si se le pasa '@psiTipoCon = 2' en el cual deberá leer un código en particular el SQL tampoco utilizará índices debido a las estadísticas guardadas en la ejecución anterior. Para esto se utilizará en el procedimiento almacenado la cláusula "With Recompile".

- Todo script ejecutado en una BD de producción, deberá tener la cantidad de registros procesados en las sentencias SELECT, UPDATE, INSERT y DELETE

 SAT <i>Servicio de Administración Tributaria de Lima</i>	Tipo: DOCUMENTO TÉCNICO	Código: GIN-AGS-DT003 Versión: 01
	Título: ESTÁNDARES PARA EL DESARROLLO DE SISTEMAS EN SQL	Fecha de Vigencia: 16/05/2014
	Página: 29 de 37	

y el tiempo aproximado que demora en ejecutar.

- Evitar las sentencias UNION, tratar las sentencias individualmente, como por ejemplo, ir almacenando en un temporal la información en cada sentencia.
- Para optimizar las sentencias T-SQL, se deben utilizar los siguientes recursos:
 - El Modelo de la Base de Datos, con esto se podrá conocer el diseño de las tablas relacionadas como así de sus relaciones y de su lógica.
 - Los Planes de Ejecución, para poder identificar los puntos críticos de las consultas.
 - **Las trazas, para conocer las sentencias y parámetros relacionados.**
- Para la ejecución de sentencias parametrizadas SQL incluidas en código VB6, .Net u otro lenguaje o tecnología se deberá utilizar necesariamente objetos tipo Parámetro para enviar los valores que necesita dichas sentencias para su ejecución; por ejemplo, en el caso de VB6 usa debe utilizar la colección ADODB.Parameters y en el caso de .Net se puede utilizar la clase SqlParameter u otra que realice esta función; esto dependerá de la versión del lenguaje o tecnología utilizada.
- **Las tablas temporales físicas, con prefijo “Tmp”, no deben ser usadas directamente desde algún procesos de una aplicación, ya que el fin de estas es de apoyar en los procesos masivos almacenando data temporal, por lo cual, esta información no debe ser usada en ninguna consulta, actualización, eliminación o inserción directamente dentro de una aplicativo.**
- **Evitar el uso de instrucciones T-SQL generados dinámicamente. Si esto no se puede evitar se debe hacer uso de la instrucción “sp_executesql” para la ejecución de estas cadenas.**

1.4. NORMAS REFERENTES AL USO DE MINÚSCULAS Y MAYÚSCULAS

1.4.1. ESTILOS DE GRAFÍA



Las condiciones siguientes describen distintas maneras de usar mayúsculas y minúsculas de los identificadores.

- **Grafiá Pascal:** La primera letra del identificador y la primera letra de las siguientes palabras concatenadas están en mayúsculas. El estilo de mayúsculas y minúsculas Pascal se puede utilizar en identificadores de tres o más caracteres. Por ejemplo: BackColor
- **Grafiá Camel:** La primera letra o letras del identificador está en minúscula y la primera letra de las siguientes palabras concatenadas en mayúscula. Por ejemplo: vApePaterno.

 SAT <i>Servicio de Administración Tributaria de Lima</i>	Tipo: DOCUMENTO TÉCNICO	Código: GIN-AGS-DT003 Versión: 01
	Título: ESTÁNDARES PARA EL DESARROLLO DE SISTEMAS EN SQL	Fecha de Vigencia: 16/05/2014 Página: 30 de 37

- **Mayúsculas:** Todas las letras del identificador van en mayúsculas. Por ejemplo: IO, DNI, RUC, etc.

1.4.2. REGLAS DE USO DE MAYÚSCULAS Y MINÚSCULAS PARA LOS IDENTIFICADORES

Cuando un identificador está compuesto de varias palabras, no utilice separadores, como guiones de subrayado ("_") ni guiones ("-"), entre las palabras. En su lugar, utilice la grafía correspondiente para señalar el principio de cada palabra.

Las instrucciones siguientes proporcionan reglas generales para los identificadores que se deberán seguir en los aplicativos generados.

Utilice la grafía Pascal para todos los nombres de miembros públicos, tipos y espacios de nombres que constan de varias palabras.

Tenga en cuenta que esta regla no se aplica a las instancias de campos. Por los motivos que se detallan en Instrucciones de diseño de miembros, no debería utilizar campos de instancia públicos. Haga un uso combinado de mayúsculas y minúsculas tipo Camel para los nombres de parámetros.

1.5. ANÁLISIS DE CÓDIGO

En la construcción de un objeto de base de datos se debe seguir las siguientes reglas:

SQL0001 – CUMPLIMIENTO DE NOMENCLATURA DE OBJETOS.

SQL0003 – CABECERA ACTUALIZADA.

SQL0004 – CUMPLIMIENTO DE NOMENCLATURA DE PARAMETROS Y VARIABLES.

SQL0005 – NO EXISTENCIA DE CODIGO DURO.

SQL0006 – PALABRAS RESERVADAS COMIENZAN CON MAYUSCULA.

SQL0007 – REVISAR QUE LAS SENTENCIAS INSERT, UPDATE GRABEN LOS DATOS DE AUDITORIA.

SQL0008 – USO DEL BORRADO LOGICO Y NO FISICO.

SQL0009 – MENCIÓN EXPLICITA DE CADA CAMPO CUANDO SE USA LA SENTENCIA INSERT O UPDATE.

SQL0010 – ELIMINACIÓN DE TABLAS TEMPORALES EN EL MOMENTO EN QUE SE DEJAN DE USAR.

SQL0011 – NO SE DEBEN USAR CONSULTAS TOTALES.

SR0001: Evite la instrucción **SELECT *** en procedimientos almacenados, vistas y funciones con valores de tablas
<http://msdn.microsoft.com/es-es/library/vstudio/dd193296%28v=vs.100%29.aspx>



 SAT <i>Servicio de Administración Tributaria de Lima</i>	Tipo: DOCUMENTO TÉCNICO	Código: GIN-AGS-DT003 Versión: 01
	Título: ESTÁNDARES PARA EL DESARROLLO DE SISTEMAS EN SQL	Fecha de Vigencia: 16/05/2014 Página: 31 de 37

SR0004: Evitar utilizar columnas que no tienen índices como expresiones de comprobación en predicados IN
<http://msdn.microsoft.com/es-es/library/vstudio/dd193249%28v=vs.100%29.aspx>

SR0005: Evite usar modelos que comiencen con "%" en predicados LIKE
<http://msdn.microsoft.com/es-es/library/vstudio/dd193273%28v=vs.100%29.aspx>

SR0006: Mover una referencia de columna a un lado de un operador de comparación para utilizar un índice de columna
<http://msdn.microsoft.com/es-es/library/vstudio/dd193264%28v=vs.100%29.aspx>

SR0007: Use ISNULL(columna, valor predeterminado) en columnas que admiten valores NULL en expresiones
<http://msdn.microsoft.com/es-es/library/vstudio/dd193267%28v=vs.100%29.aspx>

SR0008: Usar SCOPE_IDENTITY en lugar de @@IDENTITY
<http://msdn.microsoft.com/es-es/library/vstudio/dd172121%28v=vs.100%29.aspx>

SR0009: Evite usar tipos de longitud variable cuyo tamaño sea 1 ó 2
<http://msdn.microsoft.com/es-es/library/vstudio/dd193263%28v=vs.100%29.aspx>

SR0010: Evite usar una sintaxis desusada al unir tablas o vistas
<http://msdn.microsoft.com/es-es/library/vstudio/dd172122%28v=vs.100%29.aspx>

SR0011: Evitar usar caracteres especial en nombres de objeto
<http://msdn.microsoft.com/es-es/library/vstudio/dd172134%28v=vs.100%29.aspx>

SR0012: Evite utilizar palabras reservadas para nombres de tipo
<http://msdn.microsoft.com/es-es/library/vstudio/dd193421%28v=vs.100%29.aspx>

SR0013: El parámetro de salida (parámetro) no se rellena en todas las rutas de acceso del código
<http://msdn.microsoft.com/es-es/library/vstudio/dd172136%28v=vs.100%29.aspx>

SR0014: Se pueden perder datos cuando se convierte de {Tipo1} a {Tipo2}
<http://msdn.microsoft.com/es-es/library/vstudio/dd193269%28v=vs.100%29.aspx>

SR0015: Extraer las llamadas de función deterministas de predicados WHERE
<http://msdn.microsoft.com/es-es/library/vstudio/dd193285%28v=vs.100%29.aspx>

2. AUDITORIA



Las Pistas de Auditoria tendrán un alcance para todas las bases de datos creadas para los diversos sistemas desarrollados por la Gerencia de Informática del SAT y de terceros. El objetivo es el registro cronológico de todas las incidencias (modificaciones) que se vayan a realizar en las bases de datos, contenido información como fecha, hora, usuario, terminal y dato modificado. Se rigen bajo tres criterios:

2.1. TABLAS AUDITABLES CON HISTÓRICO

Se aplican al registro completo de las tablas donde los sistemas de información realizan transacciones de actualización de datos en algún registro. Estas tablas contarán

 SAT <i>Servicio de Administración Tributaria de Lima</i>	Tipo: DOCUMENTO TÉCNICO	Código: GIN-AGS-DT003 Versión: 01
	Título: ESTÁNDARES PARA EL DESARROLLO DE SISTEMAS EN SQL	Fecha de Vigencia: 16/05/2014
		Página: 32 de 37

obligatoriamente con los desencadenadores (Triggers) o código que permitan dejar las pistas de auditoría en las tablas creadas para tal fin; no hay excepción a esta regla. Las tablas que se clasifiquen con histórico contemplan de manera obligatoria los siguientes campos:

- Al inicio de la tabla:
 - Número de Operación (iNumOpe), el cual debe ser un campo identidad.
 - Código del Tipo del Motivo de la Operación (siCodTOp), el cual debe indicarse si es por creación, modificación o eliminación, este último solo en casos excepcionales.
 - Tipo de documento que autoriza el cambio (siTipDoc).
 - Número de referencia del documento que autoriza el cambio (cNumRef).
 - **Código de Usuario de la Operación** (siUsuOpe).
 - Fecha y Hora de **la Operación** (sdFecOpe) este debe tener por defecto la fecha actual del servidor (Getdate()). **Esta fecha siempre debe ser del servidor de base de datos al momento de realizar la operación.**
 - **Nombre del Terminal de la Operación** (cTerOpe) el cual debe tener por defecto una cadena vacía (""). **Este debe ser el nombre del equipo en la red o en su defecto la IP.**
- Al final de la tabla: Todos los campos **sin excepción** de la tabla original.

Los constraint (FK, IX, CKC, CKT, DF, etc.) que **tenga** la tabla original no se deben utilizar para el histórico.

Para tal fin se creará una Tabla Histórica de Movimientos (Auditoria) generada por un Desencadenador (Trigger), el cual genera el histórico de auditoría. Esta tabla tendrá el mismo nombre que la tabla original **más** la letra "H" **en mayúsculas**.

Por ejemplo:

Tabla principal: RDMAeDeu

Tabla Histórica: RDMAeDeuH

Para el caso de los desencadenadores no debe tener consultas ni lógica **alguna** para el fin creado.

2.2. TABLAS AUDITABLES SIN HISTÓRICO



Se aplica sobre aquellas tablas de los sistemas de información donde solamente se realicen transacciones de inserción de datos. La excepción a esta regla será la actualización (ejecución de sentencias Delete o Update) sobre los registros de la base de datos que siempre será a través de scripts de ejecución ya sea estático o programado (Scripts, Jobs ó DTS), es decir, el sistema no tendrá esta funcionalidad directamente. Estas actualizaciones se reflejarán en las tablas de auditoría correspondientes.

Las tablas que se clasifican "Sin Histórico" deben contemplar de manera obligatoria los

 SAT <i>Servicio de Administración Tributaria de Lima</i>	Tipo: DOCUMENTO TÉCNICO	Código: GIN-AGS-DT003 Versión: 01
	Título: ESTÁNDARES PARA EL DESARROLLO DE SISTEMAS EN SQL	Fecha de Vigencia: 16/05/2014 Página: 33 de 37

siguientes campos:

- **Código de Usuario** de actualización (*siCodUsu*).
- Fecha y Hora de actualización (*sdFecAct*), el cual debe tener por defecto la fecha actual del servidor (*Getdate()*). **Esta fecha siempre debe ser del servidor de base de datos al momento de realizar la operación.**
- Nombre del Terminal de actualización (*cNomTer*), el cual debe tener por defecto una cadena vacía (""). **Este debe ser el nombre del equipo en la red o en su defecto la IP.**

2.3. TABLAS AUDITABLES POR SEGURIDAD

Para las tablas cuyo registro sea modificado y requieran de una autorización especial, se debe contemplar un campo especial donde se almacena el código de usuario quien autoriza dicho cambio. La identificación de las tablas será a través de la seguridad cuando necesiten acceso a nivel de súper usuario.

Las tablas en mención deben contemplar de manera obligatoria los siguientes campos:

- **Código de Usuario que Autoriza** (*siCodUsuA*), **solo si es necesario.**
- **Código de Usuario** de actualización (*siCodUsu*)
- Fecha y Hora de actualización (*sdFecAct*), el cual debe tener por defecto la fecha actual del servidor (*Getdate()*). **Esta fecha siempre debe ser del servidor de base de datos al momento de realizar la operación.**
- Nombre del Terminal de actualización (*cNomTer*), el cual debe tener por defecto una cadena vacía (""). **Este debe ser el nombre del equipo en la red o en su defecto la IP.**

2.4. CAMPOS ADICIONALES PARA AUDITORÍA

Si el negocio necesita saber cuándo, en donde y quien creo un registro, se debe usar los siguientes campos:

- **Código de Usuario de Creación** (*siCodUsuCreacion*).
- **Fecha y Hora de Creación** (*sdFecCreacion*), el cual debe tener por defecto la fecha actual del servidor (*Getdate()*). **Esta fecha siempre debe ser del servidor de base de datos al momento de realizar la operación.**
- **Nombre del Terminal de Creación** (*cNomTerCreacion*), el cual debe tener por defecto una cadena vacía (""). **Este debe ser el nombre del equipo en la red o en su defecto la IP.**

Estos campos no deben ser actualizados en posteriores cambios del registro. Asimismo, estos campos deben estar ubicados antes de los campos de auditoría, de la siguiente manera:

```

siCodUsuCreacion smallint      not null,
sdFecCreacion   smalldatetime not null constraint DF_TABLA_SDFECCREACION default getdate(),
cNomTerCreacion char(20)       not null constraint DF_TABLA_SDNOMTERCREACION default '',
siCodUsu        smallint      not null,
sdFecAct        smalldatetime not null constraint DF_TABLA_SDfecACT default getdate(),
cNomTer         char(20)       not null constraint DF_TABLA_CNOMTER default ''

```



SAT <small>Servicio de Administración Tributaria de Lima</small>	Tipo: DOCUMENTO TÉCNICO	Código: GIN-AGS-DT003 Versión: 01
	Título: ESTÁNDARES PARA EL DESARROLLO DE SISTEMAS EN SQL	Fecha de Vigencia: 16/05/2014
	Página: 34 de 37	

2.5. CONSIDERACIONES DEL USO DE LOS CAMPOS DE AUDITORÍA

Para el uso correcto de los campos de auditoría se debe tener en cuenta lo siguiente:

- *Todos los campos de auditoría deben ser los últimos campos de la tabla. Para esta regla se tiene dos excepciones:*
 - *Una tabla histórica, el cual los campos de auditoría están al inicio.*
 - *Una tabla replicada, que al adicionar campos estos pueden ir al final de los campos de auditoría.*
- *Todos los campos de auditoría no debe permitir nulos.*
- *Todo nuevo registro debe tener los campos de auditoría llenados con la información del momento de la inserción.*
- *Toda actualización de un registro se debe actualizar todos los campos de auditoría.*
- *Los campos de auditoría no deben ser usados para consulta del negocio.*

3. BIBLIOGRAFÍA

Documento Técnico: GIN-DDS-DT001 MANUAL DE ESTÁNDARES PARA EL DESARROLLO DE SISTEMAS – V02

Actualización de los Libros en pantalla de SQL Server 2000 - junio de 2007

Libros en pantalla de SQL Server 2008 R2

Analizar el código de base de datos para mejorar la calidad del código
[\(http://msdn.microsoft.com/es-es/library/vstudio/dd172133%28v=vs.100%29.aspx\)](http://msdn.microsoft.com/es-es/library/vstudio/dd172133%28v=vs.100%29.aspx)

4. ANEXOS

4.1. ANEXO 1: PREFIJO DEL MÓDULO

La lista de módulos del sistema:

Módulos	Prefijos No Tributario (*)	Prefijos Tributario (*)	Otros (*)
Control y Cobranza	CC	TCC	
Control de Operaciones	CO	TCO	
Módulo de Consulta Geográfica		TFG	
Fiscalización		TFI	
Fraccionamiento	FR	TFR	
Gestión Documentaria	GD**	TGD**	
Interconexión Bancaria	IX		
Registro y Control de Licencias	LC		
Medidas Cautelares	MC	TMC	
Notificaciones	NO	TNO	
Persona	RD		
Registro y Determinación de Deuda	RD	TRD	
Recaudación	RE		



 SAT <i>Servicio de Administración Tributaria de Lima</i>	Tipo: DOCUMENTO TÉCNICO	Código: GIN-AGS-DT003 Versión: 01
	Título: ESTÁNDARES PARA EL DESARROLLO DE SISTEMAS EN SQL	Fecha de Vigencia: 16/05/2014
		Página: 35 de 37

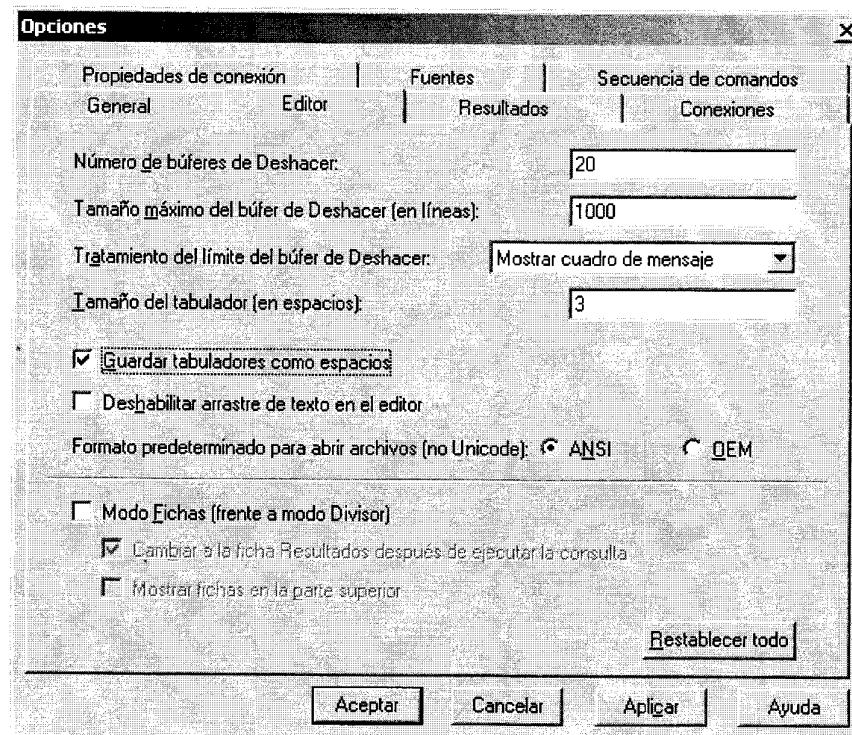
Seguridad	SE		
Sistema General	SG	TSG	
Medidas Cautelares	MC	TMC	
Sistema de Gestión Documentaria	SGD	SGD	
Remate de Bienes		RB	
Sistema de Información Geográfica (Geographic Information System)			GIS
Central Telefónica			CT
Expedientes Judiciales			EJ
Gestión de Archivo		GA	
Gestión de Cobranza	GC	TGC	

(*) El prefijo debe ser en mayúscula.

(**) De baja

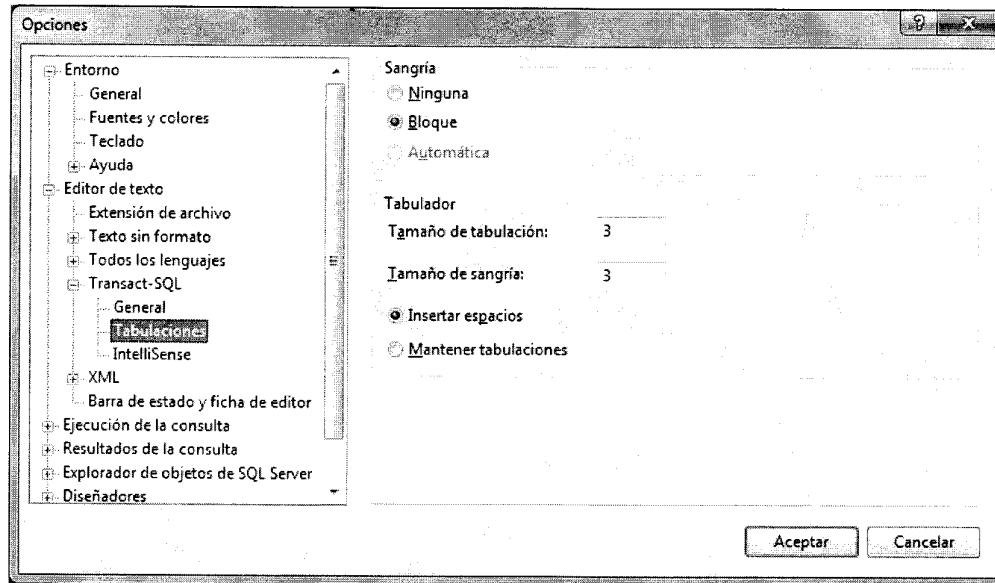
4.2. ANEXO 2: CONFIGURACIONES DE EDITORES

A continuación como configurar los espacios en los distintos editores de T-SQL:



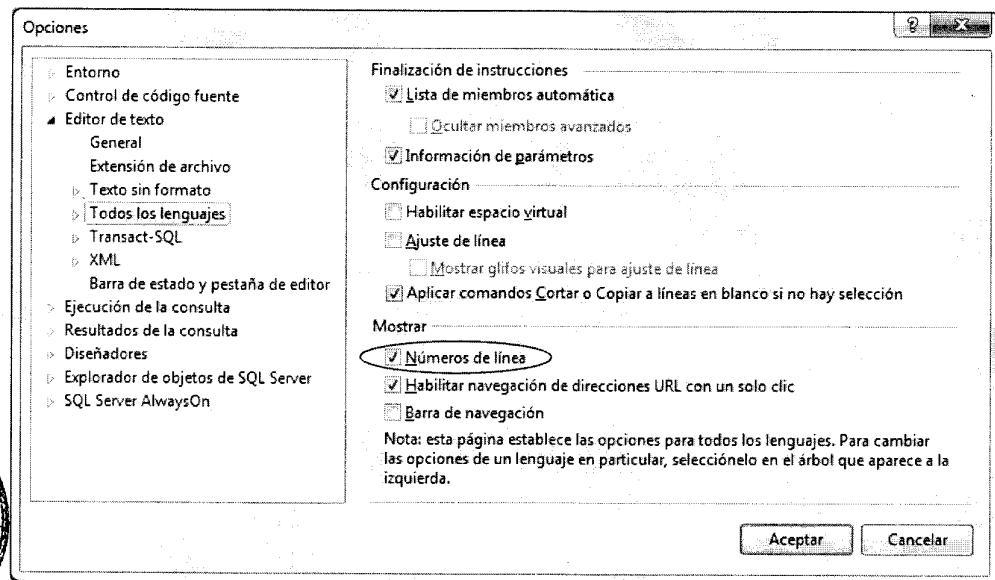
Analizador de consultas SQL (2000)

 SAT <i>Servicio de Administración Tributaria de Lima</i>	Tipo: DOCUMENTO TÉCNICO	Código: GIN-AGS-DT003 Versión: 01
	Título: ESTÁNDARES PARA EL DESARROLLO DE SISTEMAS EN SQL	Fecha de Vigencia: 16/05/2014
	Página: 36 de 37	



SQL Server Management Studio Express 2008 R2 SP2 / 2012 SP1

Se recomienda configurar el número de linea en el editor de T-SQL, para lo cual se debe activar el "Números de linea", tal como lo muestra la siguiente pantalla:



SQL Server Management Studio Express 2008 R2 SP2 / 2012 SP1

4.3. ANEXO 3: MODELO DE DATOS DEL SIAT

Para un modelo de datos del SIAT se debe considerar lo siguiente:

- **El primer campo que deben tener una tabla de tipo Maestro, Detalle o**

 SAT <i>Servicio de Administración Tributaria de Lima</i>	Tipo: DOCUMENTO TÉCNICO	Código: GIN-AGS-DT003 Versión: 01
	Título: ESTÁNDARES PARA EL DESARROLLO DE SISTEMAS EN SQL	Fecha de Vigencia: 16/05/2014 Página: 37 de 37

Movimiento debe ser el Código de Municipalidad (siCodMun).

- *El PK de la tabla debe ser compuesto por el Código de Municipalidad (siCodMun) en primer lugar.*
- *Los campos que debe tener una tabla que genera un documento definido en el Manual Interno son:*
 - *Tipo de Documento (siTipDoc)*
 - *Código de Formato del Documento (siCodFor)*
 - *Código de Unidad Organizacional (siCodUOr)*
- *En la creación de índices se debe evaluar la necesidad de adicionar el campo Código de Municipalidad (siCodMun).*

