



SISTEMA DE VENTAS SISVEN

ESTÁNDAR DE BASE DE DATOS Arquitectura de Microservicios



ÍNDICE

1. OBJETIVO.....	3
2. ALCANCE.....	3
3. DEFINICIONES Y ABREVIATURAS	3
3.1. Definiciones	3
3.2. Abreviaturas.....	4
4. DESARROLLO	5
4.1. Nomenclatura por tipo de motor de base de datos	5
5. ESTÁNDARES DE NOMENCLATURA DE OBJETOS	6
5.1. Regla General de Escritura	7
5.2. Nombre de la Base de Datos	7
5.3. SCHEMA (Esquema).....	8
5.4. TABLE (Tabla)	8
5.5. COLUMN (Columna / Campo).....	10
5.6. CONSTRAINT (Restriccion).....	11
5.7. INDEX (Índice).....	11
5.8. VIEW (Vista)	12
5.9. STORED PROCEDURE (Procedimiento Almacenado)	12
5.10. FUNCTION (Función)	12
5.11. TRIGGER.....	12
5.12. SEQUENCE (Secuencia).....	13
5.13. CURSOR	13
5.14. TEMPORARY TABLE (Tabla Temporal)	13
5.15. VARIABLE	13
5.16. USER (Usuario de Base de Datos).....	13
7. SEGURIDAD Y CONTROL DE ACCESO	15
7.1. Principio de Mínimo Privilegio	15
7.2. Aislamiento entre Microservicios	15
7.3. Usuarios por Microservicio.....	15
7.4. Reglas de Contraseña	16
8. RESPALDO Y RECUPERACIÓN	16
8.1. Tipos de Respaldo.....	16
8.2. Reglas de Respaldo	16
8.3. Procedimiento de Restauración	16
9. CONTROL DE CAMBIOS.....	17

1. OBJETIVO

Establecer y estandarizar las reglas, convenciones y buenas prácticas que deben seguir todos los desarrolladores y especialistas de base de datos al momento de diseñar, crear o modificar objetos de base de datos dentro del sistema de ventas SISVEN.

Este estándar aplica a todos los motores de base de datos utilizados en la arquitectura de microservicios del sistema, garantizando que los objetos sean consistentes, legibles y mantenibles independientemente del motor o microservicio en el que se encuentren.

2. ALCANCE

Este documento es de cumplimiento obligatorio para todos los desarrolladores asignados a los microservicios del sistema SISVEN, así como para cualquier especialista de base de datos que intervenga en el diseño, creación o modificación de objetos de base de datos.

El estándar aplica a los siguientes microservicios y sus motores de base de datos:

N	Microservicio	Acrónimo	Motor de Base de Datos	Observación
1	Clientes	CLI	PostgreSQL v18	Motor principal del microservicio
2	Caja	CAJ	MariaDB v12.3.2 (principal)	Adicionalmente tiene MySQL 8.4.3 vía Laragon
3	Ventas	VEN	MariaDB v11.4	Motor principal del microservicio
4	Almacén	ALM	MariaDB v11.8	Motor principal del microservicio
5	Facturación	FAC	MariaDB v11.4	Motor principal del microservicio
6	Administración	ADM	MariaDB v11.4	Motor principal del microservicio

3. DEFINICIONES Y ABREVIATURAS

3.1. Definiciones

N	Termino	Definición
1	Base de Datos Relacional (BDR)	Base de datos que almacena información en tablas relacionadas entre sí mediante claves. Sigue el modelo relacional y utiliza SQL como lenguaje de consulta.
2	Base de Datos No Relacional (BDNR)	Base de datos que almacena datos en formatos distintos a tablas, como documentos, pares clave-valor o grafos. No requiere un esquema fijo.
3	Microservicio	Componente independiente del sistema SISVEN que encapsula una funcionalidad de negocio específica y posee su propia base de datos.

4	Esquema (Schema)	Agrupación lógica de objetos dentro de una base de datos. En SISVEN se utilizan dos tipos: esquema público y esquema con nombre propio.
5	Objeto de Base de Datos	Cualquier elemento creado dentro de una base de datos: tablas, columnas, índices, vistas, triggers, stored procedures, funciones, sequences, entre otros.
6	Nomenclatura	Conjunto de reglas que define como se deben nombrar los objetos de base de datos para garantizar consistencia, legibilidad y mantenibilidad.
7	Llave Primaria (PK)	Columna o conjunto de columnas que identifica de forma única cada registro de una tabla.
8	Llave Foránea (FK)	Columna o conjunto de columnas que establece una relación entre dos tablas, referenciando la llave primaria de otra tabla.
9	Estándar SQL ANSI	Especificaciones del American National Standards Institute para el lenguaje SQL, que garantiza portabilidad y compatibilidad entre diferentes motores de base de datos.
10	CamelCase	Estilo de escritura donde cada palabra inicia con mayúscula sin separadores. Ejemplo: DetallePago. En SISVEN se evita este estilo en favor del guion bajo.
11	Snake_case	Estilo de escritura donde las palabras se separan con guion bajo y se usan mayúsculas. Ejemplo: DETALLE_PAGO. Es el estilo adoptado en SISVEN.

3.2. Abreviaturas

N	Abreviatura	Definición
1	SISVEN	Sistema de Ventas
2	BDR	Base de Datos Relacional
3	BDNR	Base de Datos No Relacional
4	PK	Primary Key - Llave Primaria
5	FK	Foreign Key - Llave Foránea
6	UK	Unique Key - Restricción de Unicidad
7	CK	Check Constraint - Restricción de Verificación
8	SP	Stored Procedure - Procedimiento Almacenado
9	SH	Schema - Esquema de base de datos
10	CLI	Acrónimo del microservicio Clientes
11	CAJ	Acrónimo del microservicio Caja
12	VEN	Acrónimo del microservicio Ventas
13	ALM	Acrónimo del microservicio Almacén
14	FAC	Acrónimo del microservicio Facturación
15	ADM	Acrónimo del microservicio Administración
16	ANSI	American National Standards Institute

4. DESARROLLO

4.1. Nomenclatura por tipo de motor de base de datos

La siguiente tabla muestra que objetos de base de datos aplican a cada motor utilizado en el sistema SISVEN. Los motores en uso son PostgreSQL v18 (microservicio Clientes), MariaDB en sus versiones 11.4, 11.8 y 12.3.2 (microservicios Ventas, Almacén, Facturación, Administración y Caja), y MySQL 8.4.3 (auxiliar en microservicio Caja vía Laragon).

Motores por microservicio:

N	Microservicio	Motor Principal	Versión / Observación
1	Clientes	PostgreSQL	v18 — Motor único del microservicio
2	Caja	MariaDB	v12.3.2 principal + MySQL 8.4.3 auxiliar vía Laragon
3	Ventas	MariaDB	v11.4
4	Almacén	MariaDB	v11.8
5	Facturación	MariaDB	v11.4
6	Administración	MariaDB	v11.4

Compatibilidad de objetos por motor:

N	Objeto de BD	PostgreSQL v18	MariaDB	MySQL 8.4	Observación
1	USER (Usuario BD)	SI	SI	SI	Aplica a todos los motores de SISVEN
2	TABLE (Tabla)	SI	SI	SI	Objeto principal. Aplica a todos
3	COLUMN (Columna)	SI	SI	SI	Aplica a todos los motores
4	CONSTRAINT	SI	SI	SI	PK, FK, UK, CK en todos los motores
5	INDEX (Índice)	SI	SI	SI	Aplica a todos los motores
6	VIEW (Vista)	SI	SI	SI	Aplica a todos los motores
7	STORED PROCEDURE	SI	SI	SI	Disponible en los tres motores
8	FUNCTION (Función)	SI	SI	SI	Disponible en los tres motores
9	TRIGGER	SI	SI	SI	Disponible en los tres motores
10	SEQUENCE (Secuencia)	SI	SI	NO	MySQL no soporta SEQUENCE nativo. Usar AUTO_INCREMENT.
11	CURSOR	SI	SI	SI	Disponible en los tres motores

12	TEMPORARY TABLE	SI	SI	SI	Aplica a todos los motores
13	VARIABLE	SI	SI	SI	En funciones y procedures de todos
14	MATERIALIZED VIEW	SI	NO	NO	Solo disponible en PostgreSQL v18
15	SCHEMA (Esquema)	SI	SI	SI	sh_<microservicio> en todos los motores
16	TABLESPACE	SI	SI	SI	Aplica a todos los motores
17	DATABASE LINK (DBLINK)	NO	NO	NO	No aplica en arquitectura de microservicios SISVEN
18	PACKAGE	NO	NO	NO	Exclusivo de Oracle. No aplica en SISVEN
19	SYNONYM	NO	NO	NO	Exclusivo de Oracle. No aplica en SISVEN

Nota: SEQUENCE en MySQL: MySQL no tiene soporte nativo para SEQUENCE. En su lugar, usar columnas con AUTO_INCREMENT para llaves primarias. La nomenclatura SQ_<Tabla>_PK aplica para PostgreSQL y MariaDB. En MySQL, documentar el campo AUTO_INCREMENT usando la misma convención de columnas: n_Id_<Entidad>_PK.

Nota: MATERIALIZED VIEW: Solo disponible en PostgreSQL v18 (microservicio Clientes). Para los microservicios con MariaDB o MySQL, usar VIEW normal (VW_) o tablas de resumen actualizadas mediante procedures o triggers.

Nota: MAYÚSCULAS vs minúsculas: En PostgreSQL los identificadores sin comillas se convierten automáticamente a minúsculas internamente. En MariaDB y MySQL en Linux, los nombres de tablas son sensibles a mayúsculas/minúsculas. Por esta razón, en SISVEN se adopta la convención de escribir SIEMPRE en minúsculas los nombres de objetos en MariaDB y MySQL, y en mayúsculas en PostgreSQL, salvo que el equipo decida unificar en minúsculas para todos.

5. ESTÁNDARES DE NOMENCLATURA DE OBJETOS

Esta sección define las reglas de nomenclatura para cada tipo de objeto de base de datos del sistema SISVEN. Todas las reglas se basan en el estándar SQL ANSI con referencia a las buenas prácticas de Oracle y SQL Server.

5.1. Regla General de Escritura

N	Regla	Correcto	Incorrecto
1a	En PostgreSQL: los objetos se escriben en MAYÚSCULAS.	MAE_PRODUCTO	Tab_Mae_Producto / tab_mae_producto
1b	En MariaDB y MySQL: los objetos se escriben en minúsculas	tab_mae_producto	TAB_MAE_PRODUCTO / Tab_Mae_Producto
2	Separar palabras con guion bajo (_). No usar CamelCase.	TAB_DET_DETALLE_PAGO	TabDetDetallePago / TABDETDETALLEPAGO
3	No usar la letra Ñ. Reemplazar con NI para que no suene raro.	anio / ninio	Año / niño
4	Solo caracteres A-Z, 0-9 y guion bajo (_). Sin espacios ni caracteres especiales.	FECHA_REGISTRO	fecha registro / fecha-registro
5	Nombres en singular.	TAB_MAE_CLIENTE	TAB_MAE_CLIENTES
6	Nombres en español (términos técnicos SQL en ingles son aceptados).	n_Codigo_Usuario	n_User_Code
7	Longitud máxima de 30 caracteres en la mayoría de objetos.	n_Id_Detalle_Pago	n_Identificador_Detalle_Pago_Sistema

5.2. Nombre de la Base de Datos

El nombre de la base de datos de cada microservicio debe seguir la siguiente nomenclatura:

Tipo de Nombre	Formato	Ejemplo
Nombre compuesto	BD_<Nombre_Microservicio>	BD_VENTAS_MICROSERVICIO / BD_CLIENTES_MICROSERVICIO
Nombre simple	BD_<Nombre>	BD_ALMACEN / BD_FACTURACION

Nota: El nombre de la base de datos debe estar en **singular**. Ejemplo correcto: BD_VENTA, BD_CLIENTE, BD_ALMACEN. No usar plural: BD_VENTAS, BD_CLIENTES son incorrectos.

Bases de datos del sistema SISVEN:

N	Microservicio	Nombre de BD	Motor
1	Cliente	BD_CLIENTE	PostgreSQL v18
2	Caja	BD_CAJA	MariaDB v12.3.2 / MySQL 8.4.3
3	Venta	BD_VENTA	MariaDB v11.4



4	Almacén	BD_ALMACEN	MariaDB v11.8
5	Facturación	BD_FACTURACION	MariaDB v11.4
6	Administración	BD_ADMINISTRACION	MariaDB v11.4

5.3. SCHEMA (Esquema)

Cada base de datos del sistema SISVEN utiliza dos tipos de esquemas:

Tipo de Esquema	Formato	Ejemplo	Uso
Publico	public	public	Esquema por defecto del motor. Objetos generales o de uso común.
Nombre Propio	sh_<nombre_microservicio>	sh_venta_microservicio /sh_cliente_microservicio /sh_caja	Esquema propio del microservicio. Aquí se crean todos los objetos del negocio.

Nota: Se recomienda que todos los objetos del negocio (tablas, vistas, procedimientos) se creen dentro del esquema con nombre propio (sh_ventas, sh_clientes, etc.) y NO en el esquema público. El esquema público se reserva para objetos utilitarios o de configuración del motor.

Esquemas del sistema SISVEN:

N	Microservicio	Esquema Propio	Uso
1	Cliente	sh_cliente	Tablas, vistas y objetos del dominio de clientes
2	Caja	sh_caja	Tablas, vistas y objetos del dominio de caja
3	Ventas	sh_venta	Tablas, vistas y objetos del dominio de ventas
4	Almacén	sh_almacen	Tablas, vistas y objetos del dominio de almacén
5	Facturación	sh_facturacion	Tablas, vistas y objetos del dominio de facturación
6	Administración	sh_administracion	Tablas, vistas y objetos del dominio de administración

Para referenciar un objeto siempre especificar el esquema:

Correcto	Incorrecto
sh_ventas.CAB_VENTA	CAB_VENTA (sin esquema)
sh_clientes.MAE_CLIENTE	public.MAE_CLIENTE (no usar public para tablas del negocio)

5.4. TABLE (Tabla)

El nombre de la tabla debe seguir la siguiente nomenclatura (máximo 30 caracteres):

Formato	Ejemplo 1 palabra	Ejemplo 2-3 palabras
TAB_<Tipo>_<Entidad>	TAB_MAE_USUARIO	TAB_DET_DETALLE_PAGO
TAB_<Tipo><Entidad><Correl>	TAB_MAE_USUARIO_01	TAB_DET_DETALLE_PAGO_01

Prefijos de tipo de tabla:

Prefijo	Tipo	Descripción y ejemplo
MAE	Maestra	Tablas de catálogo o configuración que raramente cambian. Eje: TAB_MAE_PRODUCTO, TAB_MAE_CLIENTE, TAB_MAE_TIPO_DOCUMENTO
CAB	Cabecera	Tabla principal de una transacción. Eje: TAB_CAB_VENTA, TAB_CAB_FACTURA, TAB_CAB_PEDIDO
DET	Detalle	Tabla de líneas o ítems asociados a una cabecera. Eje: TAB_DET_VENTA, TAB_DET_FACTURA, TAB_DET_PEDIDO
AUD	Auditoria	Tabla que registra cambios para trazabilidad. Eje: TAB_AUD_MOVIMIENTO, TAB_AUD_USUARIO
TMP	Temporal	Tabla de uso temporal para procesamiento. Eje: TAB_TMP_CALCULO_IGV, TAB_TMP_RESUMEN
PAR	Parámetros	Tabla de configuración o parámetros del sistema. Eje: TAB_PAR_PARAMETRO, TAB_PAR_CONFIG

Nota: Las tablas deben nombrarse en SINGULAR. Usar TAB_MAE_CLIENTE, no TAB_MAE_CLIENTES. Cuando el nombre tiene 2 o 3 palabras, separarlas siempre con guion bajo: TAB_DET_DETALLE_PAGO, no TABDetallePago.

Nota: El formato TAB_<Tipo>_<Entidad> permite que <Entidad> esté compuesta por una, dos o tres palabras, siempre separadas por guion bajo. Ejemplos: TAB_MAE_PRODUCTO (entidad de 1 palabra), TAB_MAE_TIPO_DOCUMENTO (entidad de 2 palabras), TAB_AUD_MOVIMIENTO_CAJA (entidad de 2 palabras).

Ejemplos reales en SISVEN:

N	Microservicio	Nombre de Tabla	Descripción
1	Venta	TAB_CAB_VENTA	Cabecera del registro de venta
2	Venta	TAB_DET_VENTA	Líneas de productos por venta
3	Cliente	TAB_MAE_CLIENTE	Catálogo maestro de clientes
4	Cliente	TAB_MAE_DOCUMENTO _TIPO	Catálogo de tipos de documento
5	Facturación	TAB_CAB_FACTURA	Cabecera del comprobante de facturación

6	Facturación	TAB_DET_FACTURA	Líneas del comprobante de facturación
7	Caja	TAB_AUD_MOVIMIENTO_CAJA	Auditoria de movimientos de caja
8	Almacén	TAB_MAE_PRODUCTO	Catalogo maestro de productos
9	Administración	TAB_PAB_PARAMETRO	Parámetros de configuración del sistema

5.5. COLUMN (Columna / Campo)

El nombre de la columna usa un prefijo que indica el tipo de dato, seguido de la descripción en español separada por guion bajo. Máximo 30 caracteres.

Formato	Uso	Ejemplo
<Prefijo>_<Descripción>	Campo normal	n_Id_Usuario / c_Nombre / d_Fecha_Venta
<Prefijo>_<Descripción>_PK	Llave Primaria	n_Id_Usuario_PK / n_Id_Venta_PK
<Prefijo>_<Descripción>	Llave Foránea (FK)	n_Id_Cliente / n_Id_Producto (sin sufijo)
<Prefijo>_<Descripción>_<Correl>	Campo con numero	c_Nombre_01 / n_Precio_02

Prefijos por tipo de dato:

Prefijo	Tipos de Dato que representa	Ejemplo de columna
c	CHAR, VARCHAR, VARCHAR2, CLOB, TEXT	c_Nombre / c_Apellido / c_Descripcion
n	INTEGER, NUMBER, NUMERIC, FLOAT, DECIMAL, BIGINT	n_Id_Usuario_PK / n_Precio / n_Cantidad
d	DATE, DATETIME, TIMESTAMP, TIME	d_Fecha_Venta / d_Fecha_Nacimiento
b	BLOB, BYTEA, BINARY, IMAGE	b_Foto_Producto / b_Archivo_Pdf

Nota: Regla importante sobre la Ñ: Cuando un nombre contiene la letra Ñ, se debe reemplazar con la secuencia NI para evitar errores en el motor y para que el nombre no suene inadecuado. Ejemplo: AÑO par ano tener errores escribimos ANIO (no AÑO), NINIO (no NIÑO), COMPANIA (no COMPAÑÍA).

Nota: No usar CHAR(N) cuando N sea mayor a 1. Usar siempre VARCHAR o TEXT para cadenas variables.

Ejemplos de columnas en SISVEN:

N	Nombre de Columna	Tipo de Dato	Descripción
---	-------------------	--------------	-------------

1	n_Id_Venta_PK	INTEGER	Identificador único de la venta (Llave Primaria)
2	n_Id_Cliente	INTEGER	Referencia al cliente (Llave Foránea)
3	c_Nombre	VARCHAR	Nombre del cliente
4	c_Apellido	VARCHAR	Apellido del cliente
5	n_Precio_Unitario	DECIMAL	Precio unitario del producto vendido
6	n_Cantidad	INTEGER	Cantidad de productos en la línea de venta
7	d_Fecha_Venta	DATE	Fecha en que se registró la venta
8	c_Anio_Fiscal	VARCHAR	Anio fiscal del registro (sin N con tilde)
9	b_Foto_Producto	BLOB	Imagen o foto del producto en almacén

5.6. CONSTRAINT (Restriccion)

Máximo 30 caracteres. Formato:

Formato	Ejemplo simple	Ejemplo con correlativo
CST_<Tabla>_<Tipo>	CST_CAB_VENTA_PK	CST_CAB_VENTA_FK_01
CST_<Tabla>_<Tipo>_<Correl>	CST_MAE_CLIENTE_UK	CST_DET_VENTA_CK_01

Código	Tipo	Descripción
PK	Primary Key	Identifica de forma única cada registro de la tabla.
FK	Foreign Key	Referencia a la llave primaria de otra tabla.
UK	Unique Key	Garantiza que los valores de una columna no se repitan.
CK	Check	Verifica que los valores cumplan una condición específica.

5.7. INDEX (Índice)

Formato	Ejemplo	Descripción
INX_<Tabla>	INX_CAB_VENTA	Índice básico sobre la tabla
INX_<Tabla>_<Tipo>	INX_CAB_VENTA_PK	Índice de tipo llave primaria
INX_<Tabla>_<Correl>	INX_MAE_PRODUCTO_01	Segundo índice sobre la misma tabla
INX_<Tabla>_<Tipo>_<Correl>	INX_MAE_CLIENTE_PK_01	Índice con tipo y correlativo

Nota: Los índices deben almacenarse en el tablespace de Índices y NO en el tablespace de Datos.

5.8. VIEW (Vista)

Formato	Ejemplo	Descripción
VW_<Descripcion_Consulta>	VW_VENTAS_DIARIAS	Vista de ventas del día
VW_<Descripcion_Consulta>	VW_CLIENTE_ACTIVO	Vista de clientes activos
VW_<Descripcion_Consulta>	VW_STOCK_ALMACEN	Vista de stock disponible en almacén

5.9. STORED PROCEDURE (Procedimiento Almacenado)

Formato	Ejemplo	Descripción
SP_<Tipo>_<Descripción>	SP_INS_VENTA	Insertar una nueva venta
SP_<Tipo>_<Descripción>_<Correl>	SP_LIS_CLIENTE_01	Primer listado de clientes
SP_<Descripción>	SP_PROCESO_CIERRE	Procedimiento sin tipo específico

Código	Operación	Descripción
INS	Insertar	Procedimiento que realiza inserciones.
ACT	Actualizar	Procedimiento que actualiza registros existentes.
ELI	Eliminar	Procedimiento que elimina registros.
BUS	Buscar	Procedimiento que busca registros específicos.
LIS	Listar	Procedimiento que retorna listados de datos.
RPT	Reporte	Procedimiento orientado a la generación de reportes.

5.10. FUNCTION (Función)

Formato	Ejemplo	Descripción
FN_<Descripción>	FN_CALCULAR_IGV	Calcula el IGV de una venta
FN_<Descripción>	FN_TOTAL_VENTA	Calcula el total de una venta
FN_<Descripción>	FN_VALIDAR_STOCK	Valida si hay stock disponible

5.11. TRIGGER

Formato	Ejemplo	Descripción
TRG_<Objeto>_<Tipo>	TRG_CAB_VENTA_INS	Trigger de inserción en tabla CAB_VENTA
TRG_<Modulo>_<Objeto>_<Tipo>	TRG_VEN_CAB_VENTA_ACT	Trigger del módulo ventas al actualizar

Tipo	Descripción
INS	Se ejecuta al insertar un nuevo registro.
ACT	Se ejecuta al actualizar un registro existente.
ELI	Se ejecuta al eliminar un registro.



5.12. SEQUENCE (Secuencia)

Formato	Ejemplo	Descripción
SQ_<Descripción>	SQ_CAB_VENTA_PK	Secuencia para PK de ventas
SQ_<Descripción>_<Correl>	SQ_MAE_CLIENTE_PK	Secuencia para PK de clientes

Nota: Si la sequence está relacionada a la llave primaria de una tabla, agregar PK en la descripción. Ejemplo: SQ_CAB_FACTURA_PK.

5.13. CURSOR

Formato	Ejemplo	Descripción
CUR_<Descripción>	CUR_VENTAS_DIA	Cursor de ventas del día
CUR_<Descripción>	CUR_STOCK_BAJO	Cursor de productos con stock bajo

5.14. TEMPORARY TABLE (Tabla Temporal)

Formato	Ejemplo	Descripción
TMP_<Descripción>	TMP_CALCULO_IGV	Tabla temporal para calcular IGV
TMP_<Descripción>_<Correl>	TMP_RESUMEN_01	Primera tabla temporal de resumen

5.15. VARIABLE

Formato	Ejemplo	Descripción
V_<Descripción>	V_Contador	Variable genérica sin tipo
V<Tipo>_<Descripción>	VI_Contador	Variable de tipo entero
V<Tipo>_<Descripción>	VC_Nombre_Cliente	Variable de tipo texto
V<Tipo>_<Descripción>	VD_Fecha_Proceso	Variable de tipo fecha
V<Tipo>_<Descripción>	VN_Total_Venta	Variable de tipo numérico

Prefijo	Tipo de dato	Ejemplo
I	INTEGER	VI_Contador
C	CHAR, VARCHAR, TEXT	VC_Nombre
D	DATE, DATETIME	VD_Fecha_Proceso
N	NUMBER, FLOAT, DECIMAL	VN_Total
R	Registro (fila)	VR_Cliente

5.16. USER (Usuario de Base de Datos)

Formato	Ejemplo	Descripción
<Acrónimo>_USR	VEN_USR	Usuario principal del microservicio Ventas
<Acrónimo>_USR_RO	CLI_USR_RO	Usuario de solo lectura del microservicio Cliente

<Acrónimo>_USR_ADM	FAC_USR_ADM	Usuario administrador del microservicio Facturación
<Acrónimo>_USR_<Correl>	CAJ_USR_01	Usuario alternativo con correlativo

6. RESTRICCIONES GENERALES

Las siguientes restricciones aplican a TODOS los objetos de base de datos sin excepción, independientemente del tipo de objeto o del motor de base de datos utilizado en el sistema SISVEN.

N	Restriccion	Detalle
1	Caracteres permitidos	Solo se permiten A-Z, 0-9 y guion bajo (_). Sin espacios, tildes, ni caracteres especiales.
2	Letra Ñ	La letra Ñ no está permitida. Reemplazar siempre con NI. Ejemplo: ANIO, NINIO, COMPANIA.
3	Idioma	Los nombres deben estar en español. Términos técnicos SQL en ingles son aceptados (INDEX, VIEW, TRIGGER, etc.).
4	Singular	Los nombres deben estar en singular. Usar MAE_CLIENTE, no MAE_CLIENTES.
5	Mayúsculas / Minúsculas	En general, todos los objetos en MAYÚSCULAS. En MySQL y MariaDB, usar minúsculas por compatibilidad del motor.
6	Sin CamelCase	No usar CamelCase. Siempre separar palabras con guion bajo. Correcto: DETALLE_PAGO. Incorrecto: DetallePago.
7	Sin comillas	No se deben usar comillas en la definición del nombre de ningún objeto.
8	Longitud máxima	Respetar el límite de caracteres de cada tipo de objeto (generalmente 30). Usar acrónimos si el nombre es muy largo.
9	Sin palabras reservadas	No usar palabras reservadas SQL como nombre (TABLE, SELECT, FROM, ORDER, GROUP, INDEX, etc.).
10	Correlativo con ceros	El correlativo va a 2 dígitos con relleno de ceros a la izquierda: 01, 02, ..., 09, 10.
11	Tablespace de índices	Los índices deben almacenarse en el tablespace de Índices, NO en el tablespace de Datos.
12	Referencia con esquema	Al ejecutar scripts siempre especificar: [SCHEMA].[NOMBRE_OBJETO]. Ejemplo: sh_ventas.CAB_VENTA.
13	Aislamiento de microservicio	Un microservicio NO debe acceder directamente a la BD de otro. La comunicación es solo vía API.
14	Mayúsculas en MariaDB/MySQL	En motores MariaDB y MySQL, los nombres van en minúsculas para compatibilidad del motor en sistemas Linux.



7. SEGURIDAD Y CONTROL DE ACCESO

En los microservicios, la seguridad de la base de datos es crítica porque cada microservicio tiene su propia base de datos independiente. Las siguientes reglas garantizan que el acceso sea controlado y con el mínimo privilegio necesario.

7.1. Principio de Mínimo Privilegio

Cada microservicio debe contar con un usuario de base de datos propio que únicamente tenga los permisos estrictamente necesarios. No se deben otorgar permisos de administrador a usuarios de aplicación.

Tipo de Usuario	Permisos	Uso
Usuario de Aplicación	SELECT, INSERT, UPDATE, DELETE	Usado por la aplicación del microservicio para operar.
Usuario de Lectura (_RO)	SELECT únicamente	Para reportes, monitoreo o consultas de solo lectura.
Usuario Administrador (_ADM)	Todos los permisos	Uso exclusivo del administrador de base de datos (DBA). NUNCA debe ser usado por la aplicación.

7.2. Aislamiento entre Microservicios

Nota: Esta estrictamente PROHIBIDO que el usuario de un microservicio tenga acceso a la base de datos de otro microservicio. La comunicación entre microservicios se realiza EXCLUSIVAMENTE a través de sus APIs, nunca mediante acceso directo a base de datos.

Permitido	Prohibido
VEN_USR accede a tablas de BD_VENTAS.	VEN_USR accede a tablas de BD_CLIENTES.
Microservicio Ventas consulta datos de Clientes vía API REST.	Microservicio Ventas hace JOIN directo a tabla de BD_CLIENTES.

7.3. Usuarios por Microservicio

Microservicio	Usuario Aplicación	Usuario Lectura	Usuario Admin
Cliente	CLI_USR	CLI_USR_RO	CLI_USR_ADM
Caja	CAJ_USR	CAJ_USR_RO	CAJ_USR_ADM
Venta	VEN_USR	VEN_USR_RO	VEN_USR_ADM
Almacén	ALM_USR	ALM_USR_RO	ALM_USR_ADM
Facturación	FAC_USR	FAC_USR_RO	FAC_USR_ADM
Administración	ADM_USR	ADM_USR_RO	ADM_USR_ADM

7.4. Reglas de Contraseña

N	Regla	Descripción
1	Longitud mínima	Mínimo 12 caracteres.
2	Complejidad	Combinar mayúsculas, minúsculas, números y al menos un carácter especial.
3	No reutilización	No reutilizar las ultimas 5 contraseñas.
4	Sin credenciales en código	Usuario y contraseña nunca en código fuente. Usar variables de entorno o gestor de secretos.

8. RESPALDO Y RECUPERACIÓN

La política de respaldo garantiza la disponibilidad y recuperación de los datos ante cualquier falla técnica, error humano o incidente de seguridad. Cada microservicio debe cumplir con esta política para su propia base de datos.

8.1. Tipos de Respaldo

Tipo	Frecuencia	Retención	Descripción
Completo (Full)	Semanal (Domingo)	4 semanas	Copia completa de toda la base de datos.
Incremental	Diario	7 días	Solo los cambios desde el ultimo backup.
Transaccional (Log)	Cada 6 horas	48 horas	Logs de transacciones para recuperación puntual.

8.2. Reglas de Respaldo

N	Regla
1	Los respaldos deben almacenarse en una ubicación diferente al servidor de base de datos.
2	Cada microservicio debe tener su propio proceso de respaldo independiente.
3	Verificar los respaldos con pruebas de restauración al menos una vez al mes.
4	Los archivos de respaldo deben estar protegidos con cifrado.
5	Ante cualquier cambio estructural (ALTER TABLE, DROP, etc.) se debe hacer un respaldo previo obligatorio.

8.3. Procedimiento de Restauración

N	Acción	Responsable
1	Identificar el tipo y alcance de la falla.	DBA / Especialista BD
2	Seleccionar el backup más reciente y apropiado.	DBA / Especialista BD
3	Restaurar en ambiente de prueba para validar integridad.	DBA / Especialista BD
4	Comunicar al equipo el tiempo estimado de restauración.	DBA / Líder Técnico



5	Ejecutar restauración en ambiente productivo.	DBA / Especialista BD
6	Verificar integridad de los datos restaurados.	DBA + Desarrollador del Microservicio
7	Documentar el incidente y la solución aplicada.	DBA / Especialista BD

9. CONTROL DE CAMBIOS

Versión	Fecha	Elaborado por	Sección	Descripción del Cambio
1.0	Junio 2026	Especialista BD	Todas	Creación inicial del documento Estándar de Base de Datos SISVEN.