

# GESTION DE HUELLAS TIPO SHA256 DE BASE DE DATOS SQL SERVER

GESTION DE LA CONFIGURACION DE SOFTWARE  
PRESENTADO POR: DAVID SANTAFE ZORRILLA  
DOCENTE: HAROLD ADRIAN BOLAÑOS

VERSION 1.0



## ÍNDICE

Índice.....	2
1 Descripción.....	3
2 Propósito.....	4
3 Registro de cambios.....	5
4 Introducción.....	6
5 Diagrama entidad-relación.....	7
6 Arquitectura de referencia .....	14
7 Funcionalidades .....	15
7.1 Crear una nueva conexión .....	15
7.2 Crear nuevo release y visualizar detalle.....	16
7.3 Comparar release con la versión actual.....	18
7.4 Visualizar cambios DDL .....	20
7.5 Visualizar todos los releases generados y sus cambios.....	21
8 Código fuente.....	23
9 Conclusiones .....	24
10 Recomendaciones .....	25

# 1 DESCRIPCIÓN

La auditoría de bases de datos SQL Server no es usada solamente para cumplir con requerimientos de conformidad. Se ha vuelto necesaria para el análisis de acciones de bases de datos, soluciones de problemas y la investigación de actividades sospechosas y maliciosas. La auditoría también puede ayudar a evitar acciones inapropiadas de parte de los usuarios.

Este documento describe la arquitectura del proyecto gestión de huellas tipo sha256 de base de datos SQL Server.

## 2 PROPÓSITO

Brindar una vista global de la arquitectura presentando aspectos relevantes de los componentes de software del proyecto de gestión de huellas tipo sha256 de base de datos SQL Server.

Exponer un diseño funcional con alto nivel de abstracción, asociando los componentes, relaciones, responsabilidades y alcance, para asegurar el cumplimiento de los requerimientos funcionales y la completitud de las funcionalidades definidas para el proyecto gestión de huellas tipo sha256 de base de datos SQL Server.

### 3 REGISTRO DE CAMBIOS

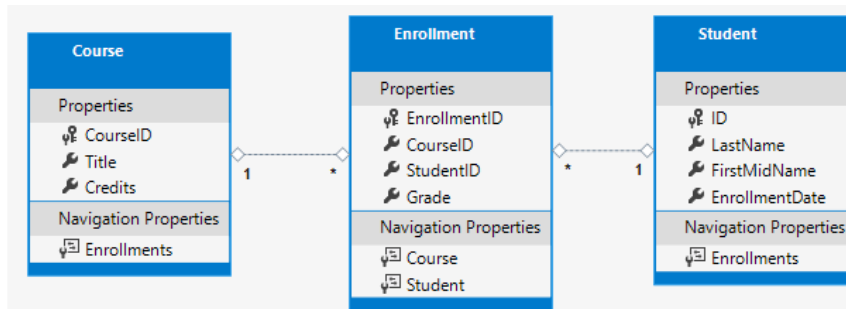
Versión	Motivo	Realizado Por	Rol	Fecha
1.0	Elaboración Inicial	David Santafe		17-05-2020

## 4 INTRODUCCIÓN

Rastrear cambios es una tarea de seguridad esencial de SQL Server. Aparte del historial de cambios, que incluye las operaciones DML (por ejemplo, INSERT, UPDATE y DELETE), el rastreo de cambios DDL en SQL Server, cambios que afectan los objetos de la base de datos, es de gran importancia también. Respecto de esto, varias técnicas pueden ser usadas como una solución de auditoría de cambios de esquema. A través de la implementación del proyecto de gestión de huellas tipo sha256 de base de datos SQL server se busca disminuir el alto porcentaje de peticiones de soporte que estén relacionados con fallos en la base de datos en detalle con alteraciones que realizan los clientes a las bases de datos y parámetros que condicionan el correcto funcionamiento del sistema.

## 5 DIAGRAMA ENTIDAD-RELACIÓN

La aplicación de muestra es un sitio web para una universidad ficticia. Incluye funcionalidades como admisión de estudiantes, creación de cursos y tareas de instructor.





Hay una relación de uno a muchos entre las entidades Student y Enrollment, y hay una relación de uno a muchos entre las entidades Course y Enrollment. En otras palabras, un estudiante puede inscribirse en cualquier número de cursos, y un curso puede tener cualquier número de estudiantes inscritos en él.

Adicionalmente las tablas necesarias para la gestión de huellas tipo sha256 de base de datos SQL Server.



AuditEventsTmp		
 Id	INT	
SchemaName	VAR CHAR(100)	
ObjectName	VAR CHAR(100)	
ObjectType	VAR CHAR(100)	
EventTime	DATETIME	
Object Key	VAR CHAR(MAX)	
Object Key Compare	VAR CHAR(MAX)	
Flag	BIT	
Release Name	VAR CHAR(50)	
Create Date	DATETIME	
Modify Date	DATETIME	
Object Id	BIGINT	

AuditEvents		
 Id	INT	
SchemaName	VAR CHAR(100)	
ObjectName	VAR CHAR(100)	
ObjectType	VAR CHAR(100)	
EventTime	DATETIME	
Object Key	VAR CHAR(MAX)	
Release Name	VAR CHAR(50)	
Create Date	DATETIME	
Modify Date	DATETIME	
Object Id	BIGINT	

AuditEventsDDL		
 Id	INT	
DDLObjectName	VAR CHAR(100)	
DDLObjectType	VAR CHAR(100)	
DDLEventTime	DATETIME	
DDLCommand	VAR CHAR(MAX)	
DDLLogin Name	VAR CHAR(150)	
DDLUserName	VAR CHAR(150)	
DDLDatabaseName	VAR CHAR(150)	
DDLSchemaName	VAR CHAR(150)	











ParametersGlobal		
 Id	INT	
Name	VAR CHAR(100)	
Value	VAR CHAR(MAX)	
Active	BIT	
Type	VAR CHAR(100)	
Create Date	DATETIME	
Modify Date	DATETIME	

Tabla: [dbo].[AuditEvents]

Descripción: Tabla encargada de almacenar los releases generados con las huellas tipo sha256 de los objetos de base de datos (tablas, procedimientos, funciones, vistas, triggers, secuencias y parámetros del sistema).

Table: AuditEvents										
Tracking/Tracking/dbo/AuditEvents										
 Info  Columns  Data  Row Count  Primary Key  Indexes  Grants  Row Id  References										
TABLE_CAT	TABLE_SCHEM	TABLE_NAME	COLUMN_NAME	DATA_TYPE	TYPE_NAME	COLUMN_SIZE	BUFFER_LENGTH	DECIMAL_DIGITS	NUM_PREC_RADIX	NULLABLE
Tracking	dbo	AuditEvents	Id	4 int identity		10	4	0	10	0
Tracking	dbo	AuditEvents	SchemaName	12 varchar		100	100	0	0	1
Tracking	dbo	AuditEvents	ObjectName	12 varchar		100	100	0	0	1
Tracking	dbo	AuditEvents	ObjectType	12 varchar		100	100	0	0	1
Tracking	dbo	AuditEvents	EventTime	93 datetime		23	16	3	0	1
Tracking	dbo	AuditEvents	ObjectKey	2005 varchar		2147483647	2147483647	0	0	1
Tracking	dbo	AuditEvents	ReleaseName	12 varchar		50	50	0	0	1
Tracking	dbo	AuditEvents	CreateDate	93 datetime		23	16	3	0	1
Tracking	dbo	AuditEvents	ModifyDate	93 datetime		23	16	3	0	1
Tracking	dbo	AuditEvents	ObjectId	-5 bigint		19	8	0	10	1

### Tabla: [dbo].[AuditEventsTmp]

Descripción: Tabla encargada de almacenar la comparación del release seleccionado con las huellas tipo sha256 de los objetos de base de datos contra los objetos actuales de la misma.

Table: AuditEventsTmp										
Tracking/Tracking/dbo/TABLE/AuditEventsTmp										
<a href="#">Info</a> <a href="#">Columns</a> <a href="#">Data</a> <a href="#">Row Count</a> <a href="#">Primary Key</a> <a href="#">Indexes</a> <a href="#">Grants</a> <a href="#">Row Id</a> <a href="#">References</a>										
TABLE_CAT	TABLE_SCHEM	TABLE_NAME	COLUMN_NAME	DATA_TYPE	TYPE_NAME	COLUMN_SIZE	BUFFER_LENGTH	DECIMAL_DIGITS	NUM_PREC_RADIX	NULLABLE
Tracking	dbo	AuditEventsTmp	Id	4 int identity		10	4	0	10	0
Tracking	dbo	AuditEventsTmp	SchemaName	12 varchar		100	100	0	0	1
Tracking	dbo	AuditEventsTmp	ObjectName	12 varchar		100	100	0	0	1
Tracking	dbo	AuditEventsTmp	ObjectType	12 varchar		100	100	0	0	1
Tracking	dbo	AuditEventsTmp	EventTime	93 datetime		23	16	3	0	1
Tracking	dbo	AuditEventsTmp	ObjectKey	2005 varchar		2147483647	2147483647	0	0	1
Tracking	dbo	AuditEventsTmp	ObjectKeyCompare	2005 varchar		2147483647	2147483647	0	0	1
Tracking	dbo	AuditEventsTmp	Flag	-7 bit		1	1	0	0	1
Tracking	dbo	AuditEventsTmp	ReleaseName	12 varchar		50	50	0	0	1
Tracking	dbo	AuditEventsTmp	CreateDate	93 datetime		23	16	3	0	1
Tracking	dbo	AuditEventsTmp	ModifyDate	93 datetime		23	16	3	0	1
Tracking	dbo	AuditEventsTmp	ObjectId	-5 bigint		19	8	0	10	1

### Tabla: [dbo].[AuditEventsDDL]

Descripción: Tabla encargada de almacenar la información devuelta por el desencadenador DDL usado para auditar operaciones CREATE, ALTER, DROP.

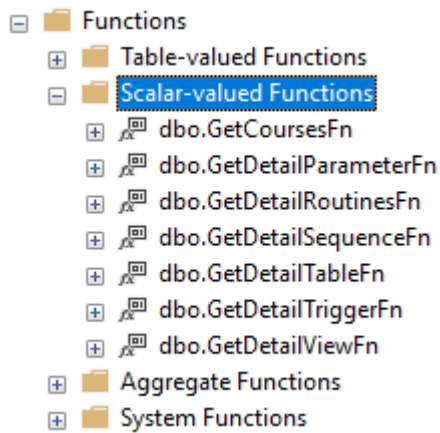
Table: AuditEventsDDL										
Tracking/Tracking/dbo/TABLE/AuditEventsDDL										
<a href="#">Info</a> <a href="#">Columns</a> <a href="#">Data</a> <a href="#">Row Count</a> <a href="#">Primary Key</a> <a href="#">Indexes</a> <a href="#">Grants</a> <a href="#">Row Id</a> <a href="#">References</a>										
TABLE_CAT	TABLE_SCHEM	TABLE_NAME	COLUMN_NAME	DATA_TYPE	TYPE_NAME	COLUMN_SIZE	BUFFER_LENGTH	DECIMAL_DIGITS	NUM_PREC_RADIX	NULLABLE
Tracking	dbo	AuditEventsDDL	Id	4 int identity		10	4	0	10	0
Tracking	dbo	AuditEventsDDL	DDLObjectName	12 varchar		100	100	0	0	1
Tracking	dbo	AuditEventsDDL	DDLObjectType	12 varchar		100	100	0	0	1
Tracking	dbo	AuditEventsDDL	DDLEventTime	93 datetime		23	16	3	0	1
Tracking	dbo	AuditEventsDDL	DDLCommand	2005 varchar		2147483647	2147483647	0	0	1
Tracking	dbo	AuditEventsDDL	DDLLoginName	12 varchar		150	150	0	0	1
Tracking	dbo	AuditEventsDDL	DDLUserName	12 varchar		150	150	0	0	1
Tracking	dbo	AuditEventsDDL	DDLDatabaseName	12 varchar		150	150	0	0	1
Tracking	dbo	AuditEventsDDL	DDLSchemaName	12 varchar		150	150	0	0	1

### Tabla: [dbo].[ParametersGlobal]

Descripción: Tabla encargada de almacenar los parámetros que condicionan el correcto funcionamiento del sistema.

Table: ParametersGlobal										
Tracking/Tracking/dbo/TABLE/ParametersGlobal										
<a href="#">Info</a> <a href="#">Columns</a> <a href="#">Data</a> <a href="#">Row Count</a> <a href="#">Primary Key</a> <a href="#">Indexes</a> <a href="#">Grants</a> <a href="#">Row Id</a> <a href="#">References</a>										
TABLE_CAT	TABLE_SCHEM	TABLE_NAME	COLUMN_NAME	DATA_TYPE	TYPE_NAME	COLUMN_SIZE	BUFFER_LENGTH	DECIMAL_DIGITS	NUM_PREC_RADIX	NULLABLE
Tracking	dbo	ParametersGlobal	Id	4 int identity		10	4	0	10	0
Tracking	dbo	ParametersGlobal	Name	12 varchar		100	100	0	0	1
Tracking	dbo	ParametersGlobal	Value	2005 varchar		2147483647	2147483647	0	0	1
Tracking	dbo	ParametersGlobal	Active	-7 bit		1	1	0	0	1
Tracking	dbo	ParametersGlobal	Type	12 varchar		100	100	0	0	1
Tracking	dbo	ParametersGlobal	CreateDate	93 datetime		23	16	3	0	1
Tracking	dbo	ParametersGlobal	ModifyDate	93 datetime		23	16	3	0	1

Funciones de base de datos:



Nombre	Descripción
[dbo].[GetDetailParameterFn]	Función encargada de mapear los parámetros del sistema para posteriormente generar la huella tipo sha256.
[dbo].[GetDetailRoutinesFn]	Función encargada de mapear los procedimientos de almacenado y funciones del sistema para posteriormente generar la huella tipo sha256.
[dbo].[GetDetailSequenceFn]	Función encargada de mapear las secuencias del sistema para posteriormente generar la huella tipo sha256.
[dbo].[GetDetailTableFn]	Función encargada de mapear las tablas del sistema para posteriormente generar la huella tipo sha256.
[dbo].[GetDetailTriggerFn]	Función encargada de mapear los desencadenadores del sistema para posteriormente generar la huella tipo sha256.
[dbo].[GetDetailViewFn]	Función encargada de mapear las vistas del sistema para posteriormente generar la huella tipo sha256.

Ejemplo: [dbo].[GetDetailTableFn]

```
BEGIN

    DECLARE @SHA VARCHAR(MAX),
            @Description VARCHAR(MAX)

    DECLARE DetailTable CURSOR FOR
    SELECT
        master.dbo.fn_varbintohexstr(HASHBYTES('SHA2_256', CONCAT(TABLE_CATALOG,
        TABLE_SCHEMA,
        TABLE_NAME,
        COLUMN_NAME,
        ORDINAL_POSITION,
        IS_NULLABLE,
        DATA_TYPE,
        CHARACTER_MAXIMUM_LENGTH,
        NUMERIC_PRECISION,
        O.CREATE_DATE,
        O.MODIFY_DATE))) SHA
    FROM SYS.SCHEMAS S
    JOIN SYS.OBJECTS O ON O.SCHEMA_ID = S.SCHEMA_ID
    JOIN INFORMATION_SCHEMA.COLUMNS C ON OBJECT_ID(CONCAT(C.TABLE_SCHEMA, '.', C.TABLE_NAME)) = O.OBJECT_ID
    WHERE O.OBJECT_ID = OBJECT_ID(@TableName);

    OPEN DetailTable
    FETCH NEXT FROM DetailTable INTO @Description

    WHILE @@fetch_status = 0

    BEGIN

        SET @SHA = CONCAT(@SHA, @Description);
        FETCH NEXT FROM DetailTable INTO @Description

    END

    CLOSE DetailTable

    DEALLOCATE DetailTable

    RETURN @SHA

END
```

## Desencadenadores de base de datos

```
CREATE TRIGGER [AuditEventsDDLTr]
ON DATABASE
FOR CREATE_TABLE, ALTER_TABLE, DROP_TABLE
AS
DECLARE
    @event xml;
SET
    @event = EVENTDATA();
INSERT INTO AuditEventsDDL
VALUES
(
    CONVERT(varchar(150),@event.query('data(/EVENT_INSTANCE/ObjectName)'),
    CONVERT(varchar(150),@event.query('data(/EVENT_INSTANCE/ObjectType)'),
    REPLACE(CONVERT(varchar(50),@event.query('data(/EVENT_INSTANCE/PostTime)'), 'T', ' '),
    CONVERT(varchar(max),@event.query('data(/EVENT_INSTANCE/TSQLCommand/CommandText)'),
    CONVERT(varchar(150),@event.query('data(/EVENT_INSTANCE/LoginName)'),
    CONVERT(varchar(150),@event.query('data(/EVENT_INSTANCE/UserName)'),
    CONVERT(varchar(150),@event.query('data(/EVENT_INSTANCE/DatabaseName)'),
    CONVERT(varchar(150),@event.query('data(/EVENT_INSTANCE/SchemaName)'));
GO

ENABLE TRIGGER [AuditEventsDDLTr] ON DATABASE
GO
```

### Desencadenador: [AuditEventsDDLTr]

Descripción: Desencadenador DDL que captura eventos CREATE, ALTER y DROP a nivel de base de datos (aunque los desencadenadores pueden ser creados a nivel del servidor para capturar eventos para todas las bases de datos en el servidor; la opción ON ALL SERVER debería usarse en lugar de ON DATABASE).

### Procedimientos de almacenado de base de datos

Nombre	Descripción
[dbo].[CreateRelease]	Procedimiento de almacenado encargado de crear el release mapeando todos los objetos (tablas, procedimientos, funciones, vistas, triggers, secuencias y parámetros del sistema) y almacenar la huella tipo sha256.
[dbo].[CompareRelease]	Procedimiento de almacenado encargado de comparar el release indicado con las huellas tipo sha256 de los objetos de base de datos contra los objetos actuales de la misma.
[dbo].[GetReleases]	Procedimiento de almacenado encargado de obtener todos los releases generados.

<b>[dbo].[GetDDLObject]</b>	Procedimiento de almacenado encargado de obtener la información devuelta por el desencadenador DDL usado para auditar operaciones CREATE, ALTER, DROP sobre un objeto TABLE seleccionado.
-----------------------------	---

Ejemplo: [dbo].[CreateRelease]

```

BEGIN

    SET @ObjectNameFull = CONCAT(@SchemaName, '.', @ObjectName)

    INSERT INTO [dbo].[AuditEvents]
        ([SchemaName]
        , [ObjectName]
        , [ObjectType]
        , [EventTime]
        , [ObjectKey]
        , [ReleaseName]
        , [CreateDate]
        , [ObjectId])
    VALUES (@SchemaName
        , @ObjectName
        , @ObjectType
        , GETDATE()
        , CASE WHEN @ObjectType = 'USER_TABLE' THEN [dbo].[GetDetailTableFn](@ObjectNameFull)
            WHEN @ObjectType IN ('SQL_STORED_PROCEDURE', 'SQL_SCALAR_FUNCTION') THEN [dbo].[GetDetailRoutinesFn](@ObjectNameFull)
            WHEN @ObjectType = 'VIEW' THEN [dbo].[GetDetailViewFn](@ObjectNameFull)
            WHEN @ObjectType = 'SEQUENCE_OBJECT' THEN [dbo].[GetDetailSequenceFn](@ObjectNameFull)
            WHEN @ObjectType = 'SQL_TRIGGER' THEN [dbo].[GetDetailTriggerFn](@ObjectName)
            WHEN @ObjectType = 'PARAMETER' THEN [dbo].[GetDetailParameterFn](@ObjectName)
        END
        , @ReleaseName
        , GETDATE()
        , @ObjectId)

    FETCH NEXT FROM ObjectsDB INTO @SchemaName, @ObjectName, @ObjectType, @ObjectId
END

CLOSE ObjectsDB

DEALLOCATE ObjectsDB

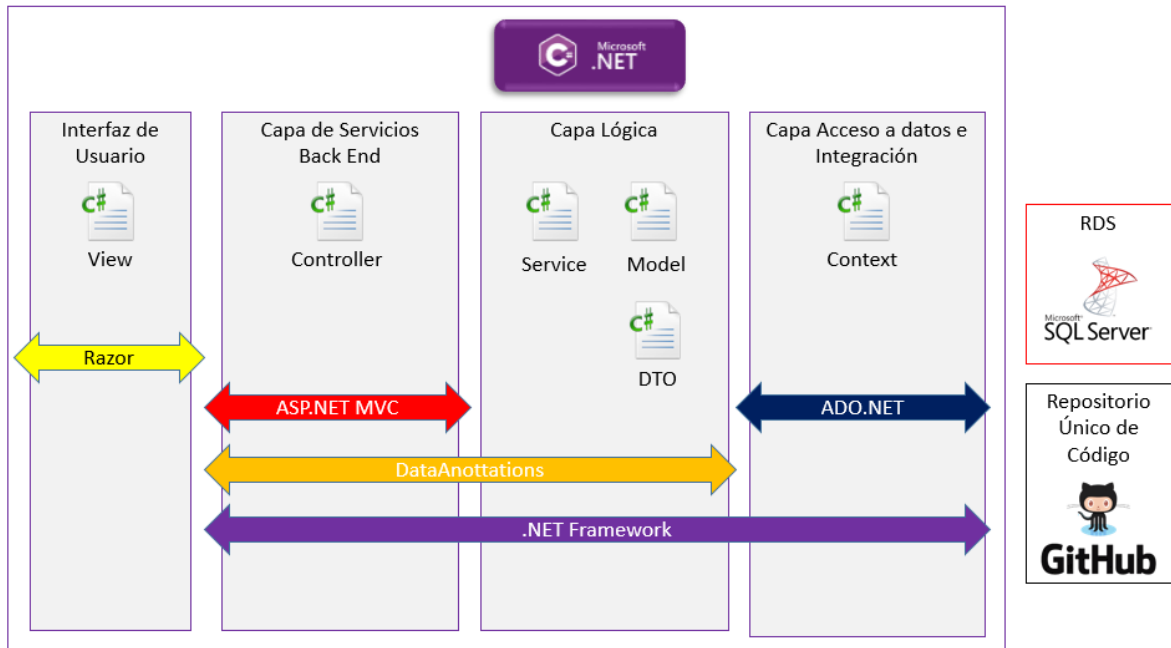
COMMIT TRAN CreateReleaseTran

SET @Message = 'The process has been executed successfully'

IF @@ERROR != 0
BEGIN
    SET @Message = ERROR_MESSAGE()
    ROLLBACK TRAN CreateReleaseTran
END

```

## 6 ARQUITECTURA DE REFERENCIA



Componente	Descripción
<b>Controlador</b>	Controla, recibe las órdenes del usuario y se encarga de solicitar los datos al modelo y de comunicárselos a la vista.
<b>Servicio</b>	Componente que contiene la lógica de procesamiento de datos aplicando validaciones de negocio.
<b>DTO</b>	Transmite información entre un cliente y un servidor, permite crear estructuras de datos independientes de nuestro modelo de datos.
<b>Contexto</b>	Contiene la lógica empresarial para la creación, recuperación, modificación y eliminación de instancias de entidad.
<b>Entidad</b>	Establece una interface común que es implementada por un proveedor de persistencia.
<b>Base de datos</b>	Conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso.

## 7 FUNCIONALIDADES

### 7.1 CREAR UNA NUEVA CONEXIÓN

Funcionalidad encargada de crear una nueva conexión a base de datos para gestión de huellas tipo sha256 de base de datos SQL Server.

La aplicación viene por defecto conectada a la base de datos a través del archivo de configuración como se muestra a continuación:

```
<connectionStrings>
  <add name="TrackingContext" connectionString="Server=DESKTOP-6844TEN\SQLEXPRESS;Initial Catalog=Tracking;Integrated Security=True"/>
</connectionStrings>
```

La conexión a base de datos implementa el patrón Singleton, por lo que al crear una nueva conexión se modifica la instancia única con la que se está trabajando.

Tracking changes in SQL Server Inicio Tracking New Connection

Create

Server

DESKTOP-6844TEN\SQLEXPRESS

The Server is required

Database

Tracking

The Database is required

User

sa

The User is required

Password

\*\*\*\*\*

The Password is required

Create

© 2020 - Tracking



## 7.2 CREAR NUEVO RELEASE Y VISUALIZAR DETALLE

Funcionalidad encargada de generar el release con las huellas tipo sha256 de los objetos de base de datos (tablas, procedimientos, funciones, vistas, triggers, secuencias y parámetros del sistema). Adicionalmente permite visualizar el detalle de los objetos incluidos y la fecha de generación.

Tracking changes in SQL Server Inicio Tracking

### Index

Create New Show History

Release Name	
Release v1	<a href="#">Details</a>   <a href="#">Compare with current</a>
Release v2	<a href="#">Details</a>   <a href="#">Compare with current</a>
Release v3	<a href="#">Details</a>   <a href="#">Compare with current</a>

© 2020 - Tracking

Tracking changes in SQL Server Inicio Tracking

### Create

Release Name

Release v4

Create

[Back to List](#)

© 2020 - Tracking

## Index

[Create New](#) | [Show History](#)

### Release Name

Release v1	<a href="#">Details</a>   <a href="#">Compare with current</a>
Release v2	<a href="#">Details</a>   <a href="#">Compare with current</a>
Release v3	<a href="#">Details</a>   <a href="#">Compare with current</a>

## Details

Schema Name	Object Name	Object Type	Event Time	Create Date	Release Name
	WorkingHours	PARAMETER	17/05/2020 23:22:05	17/05/2020 23:22:05	Release v1
	TRM	PARAMETER	17/05/2020 23:22:05	17/05/2020 23:22:05	Release v1
	Age	PARAMETER	17/05/2020 23:22:05	17/05/2020 23:22:05	Release v1
dbo	SEQ_COURSE	SEQUENCE_OBJECT	17/05/2020 23:22:04	17/05/2020 23:22:04	Release v1

## 7.3 COMPARAR RELEASE CON LA VERSIÓN ACTUAL

Funcionalidad encargada de comparar el release seleccionado con las huellas tipo sha256 de los objetos de base de datos contra los objetos actuales de la misma y visualizar los diferentes estados de los objetos de la siguiente manera:

- Objeto inicial
- Objeto eliminado
- Objeto modificado

Adicional si el objeto de la base de datos generado en el release es renombrado en el modelo de datos actual se muestra el objeto modificado y su nuevo nombre para mayor trazabilidad.

Tracking changes in SQL Server

Inicio

Tracking

Index

[Create New](#) | [Show History](#)

Release Name	
Release v1	<a href="#">Details</a> <a href="#">Compare with current</a>
Release v2	<a href="#">Details</a>   <a href="#">Compare with current</a>
Release v3	<a href="#">Details</a>   <a href="#">Compare with current</a>

© 2020 - Tracking

Tracking changes in SQL Server

Inicio

Tracking

New Connection

Classification

Object Initial

Object Deleted

Object Modify

Compare with current

Schema Name	Object Name	Object Type	Event Time	Rename	Flag	Release Name	Create Date
	WorkingHours	PARAMETER	18/05/2020 12:01:10		<input checked="" type="checkbox"/>	Release v1	18/05/2020 21:26:30
	TRM	PARAMETER	18/05/2020 12:01:10		<input type="checkbox"/>	Release v1	18/05/2020 21:26:30
	Age	PARAMETER	18/05/2020 12:01:10		<input checked="" type="checkbox"/>	Release v1	18/05/2020 21:26:30

[Back to List](#)

© 2020 - Tracking

## Classification

Object Initial

Object Deleted

Object Modify

## Compare with current

table

Schema Name	Object Name	Object Type	Event Time	Rename	Flag	Release Name	Create Date	
dbo	Course	USER_TABLE	18/05/2020 12:01:06		<input checked="" type="checkbox"/>	Release v1	18/05/2020 21:26:28	Select
dbo	Enrollment	USER_TABLE	18/05/2020 12:01:10		<input type="checkbox"/>	Release v1	18/05/2020 21:26:29	Select
dbo	ParametersGlobal	USER_TABLE	18/05/2020 12:01:10		<input type="checkbox"/>	Release v1	18/05/2020 21:26:29	Select
dbo	DocumentTypes	USER_TABLE	18/05/2020 12:01:10		<input type="checkbox"/>	Release v1	18/05/2020 21:26:30	Select
dbo	Student	USER_TABLE	18/05/2020 12:01:10		<input checked="" type="checkbox"/>	Release v1	18/05/2020 21:26:30	Select

## 7.4 VISUALIZAR CAMBIOS DDL

Funcionalidad encargada de visualizar la información almacenada por el desencadenador DDL usado para auditar operaciones CREATE, ALTER, DROP sobre un objeto TABLE seleccionado.

Tracking changes in SQL Server

Inicio

Tracking

New Connection

Object Initial

Object Deleted

Object Modify

Compare with current

table

Schema Name	Object Name	Object Type	Event Time	Rename	Flag	Release Name	Create Date	
dbo	Course	USER_TABLE	18/05/2020 12:01:06		<input checked="" type="checkbox"/>	Release v1	18/05/2020 21:26:28	Select
dbo	Enrollment	USER_TABLE	18/05/2020 12:01:10		<input type="checkbox"/>	Release v1	18/05/2020 21:26:29	Select
dbo	ParametersGlobal	USER_TABLE	18/05/2020 12:01:10		<input type="checkbox"/>	Release v1	18/05/2020 21:26:29	Select
dbo	DocumentTypes	USER_TABLE	18/05/2020 12:01:10		<input type="checkbox"/>	Release v1	18/05/2020 21:26:30	Select
dbo	Student	USER_TABLE	18/05/2020 12:01:10		<input checked="" type="checkbox"/>	Release v1	18/05/2020 21:26:30	Select

Tracking changes in SQL Server

Inicio

Tracking

New Connection

Show Object Changes

Schema Name	Object Name	Object Type	Event Time	Command	Login Name
dbo	Course	TABLE	18/05/2020 21:19:46	ALTER TABLE [dbo].[Course] ADD Duration INT	DESKTOP-6844TEN\EliteBook

Back to List

© 2020 - Tracking

## 7.5 VISUALIZAR TODOS LOS RELEASES GENERADOS Y SUS CAMBIOS

Funcionalidad encargada de visualizar todos los releases generados. Adicionalmente permite visualizar la comparación contra el release anterior visualizando los diferentes estados de los objetos de la siguiente manera:

- Objeto inicial
- Objeto nuevo
- Objeto modificado

Tracking changes in SQL Server

Inicio

Tracking

Classification

Object Initial

Object New

Object Modify

History

Release v1

Schema Name	Object Name	Object Type	Event Time
	WorkingHours	PARAMETER	18/05/2020 12:01:10
	TRM	PARAMETER	18/05/2020 12:01:10
	Age	PARAMETER	18/05/2020 12:01:10
dbo	SEQ_COURSE	SEQUENCE_OBJECT	18/05/2020 12:01:10
dbo	GetCoursesFn	SQL_SCALAR_FUNCTION	18/05/2020 12:01:10
dbo	GetCoursesSp	SQL_STORED_PROCEDURE	18/05/2020 12:01:10
	AuditEventsDDLTr	SQL_TRIGGER	18/05/2020 12:01:10
dbo	Course	USER_TABLE	18/05/2020 12:01:06
dbo	Enrollment	USER_TABLE	18/05/2020 12:01:10
dbo	ParametersGlobal	USER_TABLE	18/05/2020 12:01:10
dbo	DocumentTypes	USER_TABLE	18/05/2020 12:01:10
dbo	Student	USER_TABLE	18/05/2020 12:01:10
dbo	GetCoursesVw	VIEW	18/05/2020 12:01:10

## Release v2

Schema Name	Object Name	Object Type	Event Time
	WorkingHours	PARAMETER	18/05/2020 21:30:52
	TRM	PARAMETER	18/05/2020 21:30:52
dbo	SEQ_COURSE	SEQUENCE_OBJECT	18/05/2020 21:30:51
dbo	GetCoursesFn	SQL_SCALAR_FUNCTION	18/05/2020 21:30:51
dbo	GetStudentsSp	SQL_STORED_PROCEDURE	18/05/2020 21:30:52
dbo	GetCoursesSp	SQL_STORED_PROCEDURE	18/05/2020 21:30:50
	AuditEventsDDLTr	SQL_TRIGGER	18/05/2020 21:30:52
dbo	ParametersGlobal	USER_TABLE	18/05/2020 21:30:51
dbo	DocumentTypes	USER_TABLE	18/05/2020 21:30:51
dbo	Student	USER_TABLE	18/05/2020 21:30:51
dbo	Course	USER_TABLE	18/05/2020 21:30:50
dbo	Enrollment	USER_TABLE	18/05/2020 21:30:50
dbo	GetCoursesVw	VIEW	18/05/2020 21:30:51

## Release v3

Schema Name	Object Name	Object Type	Event Time
	WorkingHours	PARAMETER	18/05/2020 21:36:21
	TRM	PARAMETER	18/05/2020 21:36:21
dbo	SEQ_COURSE	SEQUENCE_OBJECT	18/05/2020 21:36:20
dbo	GetCoursesFn	SQL_SCALAR_FUNCTION	18/05/2020 21:36:20
dbo	GetStudentsSp	SQL_STORED_PROCEDURE	18/05/2020 21:36:21
dbo	GetCoursesSp	SQL_STORED_PROCEDURE	18/05/2020 21:36:20
	AuditEventsDDLTr	SQL_TRIGGER	18/05/2020 21:36:21
dbo	ParametersGlobal	USER_TABLE	18/05/2020 21:36:20
dbo	DocumentTypes	USER_TABLE	18/05/2020 21:36:21
dbo	Student	USER_TABLE	18/05/2020 21:36:21
dbo	Course	USER_TABLE	18/05/2020 21:36:20
dbo	Enrollment	USER_TABLE	18/05/2020 21:36:20
dbo	GetCoursesVw	VIEW	18/05/2020 21:36:20

## 8 CÓDIGO FUENTE

- Repositorio único de código: <https://github.com/dsantafe/Tracking>, código fuente del proyecto gestión de huellas tipo sha256 de base de datos SQL Server.
- Documento de referencia: DT-ARQUITECTURA-PRY-TRACKING.pdf
- DML: <https://github.com/dsantafe/Tracking/tree/master/Tracking/Schema/DML.sql>, scripts para la creación de objetos necesarios para el correcto funcionamiento del aplicativo.
- DDL: <https://github.com/dsantafe/Tracking/tree/master/Tracking/Schema/DDL.sql>, Inicializar DB con datos de prueba



## 9 CONCLUSIONES

Con la implementación de herramientas que permitan rastrear los cambios en SQL Server en este caso por medio de huellas tipo SHA256 se facilita la identificación de alteración de tablas y parámetros que condicionan el correcto funcionamiento del sistema, con el objetivo de asegurar la integridad del modelo de la base de datos y cumplir la cláusula de alteración de producto.

## 10 RECOMENDACIONES

Aunque se ha mostrado que se pueden rastrear cambios DDL usando desencadenadores y como este método puede producir resultados de auditoría viables, tiene muchas desventajas. Primero que nada, un usuario con suficientes permisos puede fácilmente deshabilitar los desencadenadores. Otra manera de evitar el rastreo de cambios DDL en SQL Server con desencadenadores es realizar un cambio de esquema y luego eliminar la información capturada del repositorio de auditoría. En nuestro caso, simplemente eliminar una entrada apropiada de la tabla [Tracking].[dbo].[AuditEventsDDL]. Además, si no existen usuarios maliciosos con privilegios que puedan realizar cambios DDL, este método no puede proveer información acerca de cambios que ocurrieron antes de que se configurara.