

SQL Collate – SQL Intercalación

El concepto de Intercalación (Collation) hace referencia al patrón de bits utilizado para representar/almacenar cada carácter, y en consecuencia también se refiere a las reglas utilizadas para ordenar y comparar caracteres. Evidentemente, se trata de un concepto que afecta sólo a los campos de texto.

El SQL Collate (referido como SQL Collation según la fuente) es la propiedad de la base de datos que proporciona la distinción entre caracteres como acentos, mayúsculas, minúsculas, otros caracteres soportados y reglas de ordenación para los datos que la base de datos es capaz de manejar.

La intercalación (SQL Collate), por tanto, especificará a la base de datos tanto los patrones de bits que representan cada carácter contenido en el juego de caracteres vinculados a una región, como las reglas de comparación u ordenación de datos.

Para entender la importancia de esto, podemos pensar en nuestra letra más castiza, la ñe (Ñ), que es una letra que no se encuentra presente en la lengua inglesa, por ejemplo. Lo mismo pasaría con el árabe, el japonés, etc.

Esto hace que no sea lo mismo utilizar un juego de caracteres que otro, ya que podemos encontrarnos con que la aplicación genere caracteres que no sean manejables o entendibles por la base de datos, comportándose de manera imprevisible.

Por todo lo expuesto anteriormente, podemos decir que el SQL Collate viene condicionado por la región donde se vaya a instalar y explotar la base de datos o el aplicativo y, tal como hemos podido ver, la configuración de este parámetro es un punto crítico del despliegue, aunque podrá ser modificado posteriormente en caso de ser necesario.

En la instalación de SQL Server se indica la **Intercalación (Collation) de la Instancia**, que definirá la intercalación utilizada por las bases de datos del sistema. De aquí podemos inferir (de hecho, es así), que todas las bases de datos del sistema utilizan la misma Intercalación (Collation).

Es importante de tener bien identificada y documentada la Intercalación utilizada por nuestras Instancias y bases de datos de SQL Server, tanto para recuperación de catástrofes como en otros procesos como migraciones y consolidaciones de servidores, etc.

Otra particularidad de la Intercalación (Collation) de una Instancia de SQL Server, es que siempre que se cree una base de datos, por defecto se creará utilizando la Intercalación de la Instancia, excepto que especifiquemos lo contrario.

Aunque la Intercalación (Collation) de todas las bases de datos del sistema debe ser la misma, **la Intercalación (Collation) de cada base de datos de usuario puede ser diferente**, es decir, podemos tener una base de datos utilizando una Intercalación por requisitos de la aplicación que la utiliza, y otra base de datos de la misma Instancia de SQL Server, utilizando una Intercalación diferente, siendo ésta una configuración natural y habitual.

Es importante tener en cuenta que **la Intercalación (Collation) de una base de datos**, sólo afecta a:

- La Intercalación de las tablas y objetos del sistema.
- La Intercalación de las nuevas tablas y objetos que se creen en la base de datos. Esto es debido, a que si no se especifica de forma explícita la Intercalación deseada en la sentencia CREATE correspondiente, se tomará por defecto la Intercalación de la base de datos.

Es posible **cambiar la Intercalación de una base de datos** utilizando el comando ALTER DATABASE COLLATE (ej: **ALTER DATABASE miBaseDatos COLLATE Modern_Spanish_CI_AS**). Sin embargo, lo más probable es que no consigamos lo que creemos estar consiguiendo (ya me decían de pequeño: "hay que tener cuidado con lo que se pide"... jeje ;-). El comando ALTER DATABASE COLLATE se limita a cambiar la Intercalación (Collation) de las tablas del sistema de la base de datos y el valor de Intercalación que por defecto utilizarán los nuevos objetos que se creen sin especificar de forma explícita la Intercalación deseada. Sin embargo, **ALTER DATABASE COLLATE no cambiará la intercalación de las tablas existentes**. Si deseamos cambiar la Intercalación de una base de datos y de todos sus objetos, una solución es crear una nueva base de datos con la Intercalación deseada, generar el Script de creación de objetos de la base de datos original (sin especificar la Intercalación en dicho Script), ejecutar el Script en la nueva base de datos, y cargar los datos desde la base de datos original utilizando el Asistente de Importación de SQL Server (Import Data). Esta no es la única opción posible, pero en general, es quizás la más apropiada (en general... en cada caso específico, habría que ver). También podemos crear a mano las descargas y cargas de las tablas, con la utilidad BCP.EXE y/o con el comando BULK INSERT, pero a fin de cuentas, esto es lo que hará el Asistente de Importación de forma automática para todas las tablas, así que para qué perder el tiempo, ¿no?. También cabe la posibilidad de utilizar el comando **ALTER TABLE ALTER COLUMN COLLATE** (ej: **ALTER TABLE Facturas ALTER COLUMN ClientIDCHAR(10) COLLATE Modern_Spanish_CI_AS**), sin embargo, en el momento en que el campo que deseamos alterar tenga definido sobre él un índice, una clave externa, etc., la ejecución de dicho ALTER TABLE será fallida, produciéndose un error como el siguiente:

```
Server: Msg 5074, Level 16, State 8, Line 1
Theobject 'PK_Facturas' isdependentoncolumn 'cod'.
Server: Msg 4922, Level 16, State 1, Line 1
ALTER TABLE ALTER COLUMN codfailedbecauseoneor more objectsaccessthiscolumn.
```

Evidentemente, eliminar los índices, claves externas, etc., y luego volver a crearlos, es un trabajo extra que tendremos que considerar si realmente nos interesa asumir.

También **es posible definir un valor de Intercalación distinto para cada campo de cada tabla**. De éste modo, podríamos tener en cada campo de texto de cada tabla, una Intercalación distinta, lo cual se trata de una posibilidad que ofrece gran flexibilidad. Sin embargo, **puede producirse un efecto colateral**: en algunas instalaciones que he conocido, me he encontrado que en una misma base de datos conviven tablas con diferentes intercalaciones, no porque se desee, sino por descuido de los programadores. Al principio no hay problema, hasta que llega ese momento en el cual deseamos hacer alguna consulta y no podemos (ej: un JOIN entre dos campos de texto con distintas intercalaciones sin utilizar COLLATE en dicho JOIN).

Para conocer las distintas Intercalaciones que tenemos disponibles, podemos consultar la función del sistema **fn_helpcollations** (ej: **select * from ::fn_helpcollations()**).

En la práctica, habitualmente al crear una base de datos se crea con la Intercalación por defecto de la Instancia (que ya vimos que se determinó en la instalación de la Instancia), de tal modo que al crear las tablas en la base de datos, sus campos de texto se crean con la Intercalación por defecto de la base de datos. Esto, siempre y cuando no se especifique de forma explícita la Intercalación en las sentencias DDL.

Otro problema habitual relacionado con la Intercalación (Collation), es cuando deseamos realizar consultas entre tablas realizando un JOIN entre campos de texto con distinta Intercalación. Esto no es posible de realizar (obtendremos un error como **Cannot resolve collation conflict for equal operation**). Sin embargo, cabe la posibilidad de utilizar la **cláusula COLLATE** en el JOIN, como si estuviésemos realizando un Casting (en vez de convertir entre tipos de datos, convertimos entre Intercalaciones). Así, podríamos utilizar una cláusula WHERE similar a la del siguiente ejemplo: **WHERE a.client_name = b.client_name COLLATE Modern_Spanish_CI_AS**.

Configurar SQL Collate en tu SQL server

Pre-requisitos o pre-configuración

Para completar de forma satisfactoria este tutorial y poder configurar SQL Collate se necesitará:

- Tener un [Servidor](#) con Windows Server.
- Tener un **SQL Server** instalado en el servidor.

Paso 1. Cómo consultar el SQL Collate de nuestra base de datos

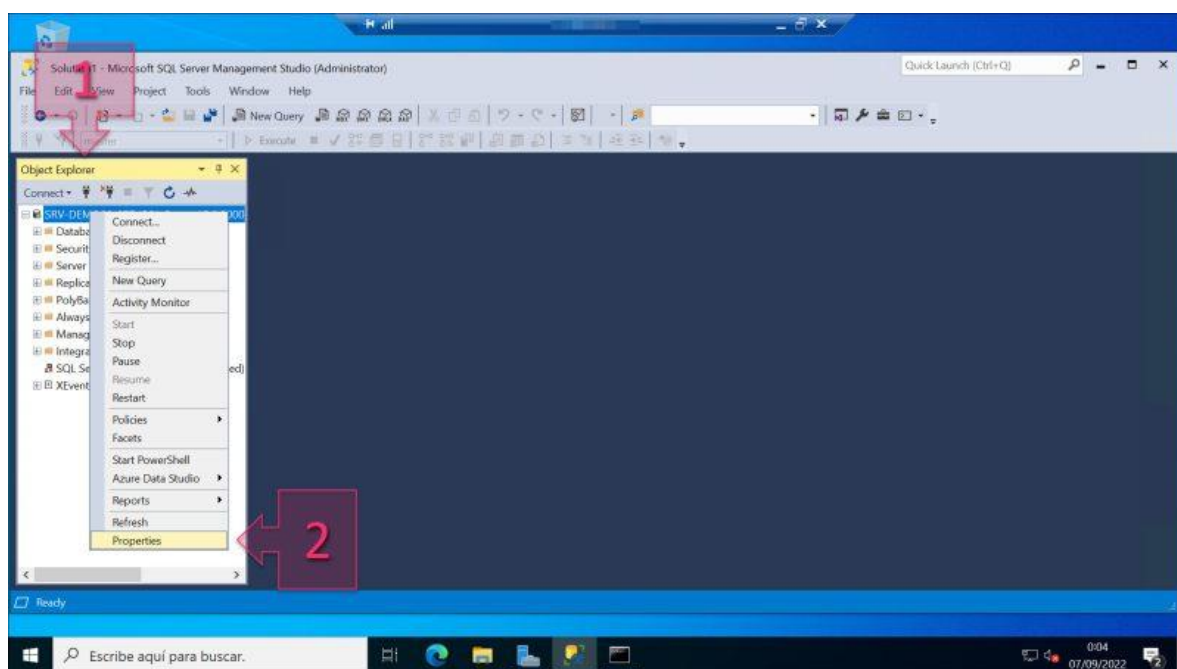
Una vez comentado esto, vamos a **ver como consular la intercalación** o SQL Collation que tenemos configurada en nuestra base de datos, para ver a continuación cómo podemos configurarlo tanto desde la consola gráfica como desde línea de comandos.

En el caso de la consulta del valor de SQL Collate, podemos realizarla de distintas maneras:

1. Podemos consultarlo desde la consola gráfica.
2. Podemos consultarlo mediante ejecución de SQL (de varias maneras).

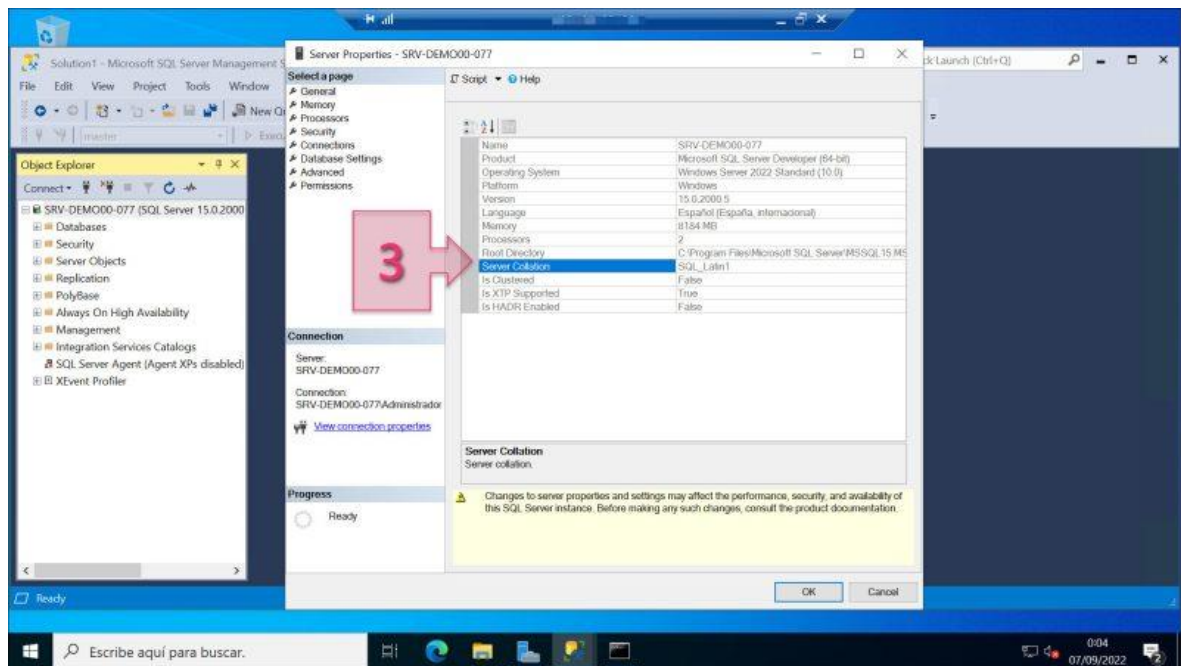
Empezaremos por ver como hacerlo de la manera más sencilla, haciendo uso de la consola gráfica de SSMS (SQL Server Management Studio), que es la herramienta de administración de SQL Server con la que todo técnico ha tenido que lidiar en alguna ocasión.

Una vez dentro de **la consola de SSMS** se debe ir al Explorador de Objetos, situado en el **lateral**, hacer clic con el **botón derecho** sobre el nombre del servidor (1) y **hacer clic en “Propiedades”** (2) entre las opciones que se muestran.



Mostramos las propiedades de la base de datos

Una vez hecho esto, se mostrará una ventana emergente con las propiedades del servidor de SQL, entre las que se debe **comprobar el valor de “Server Collation”** (3).



Comprobamos la propiedad de Server Collate dentro de las propiedades de la base de datos

Otra forma de comprobarlo es mediante una consulta de SQL. Para ello se debe ir a la consola de SSMS y hacer clic en “Nueva query” (4).

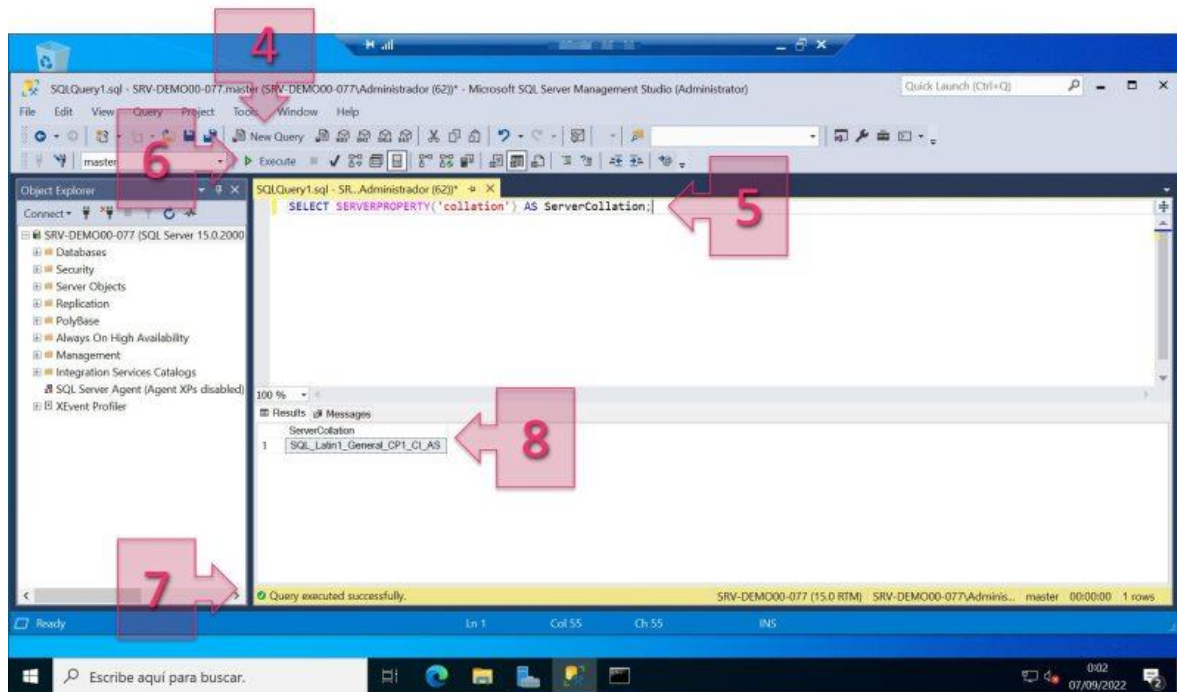
A continuación, dentro del espacio para consultas, se debe introducir la siguiente consulta (5):

```
SELECT SERVERPROPERTY('collation') AS ServerCollation;
```

Una vez hecho esto, se debe hacer clic en “Execute” (6).

Si todo va bien, se podrá ver el mensaje “Queryexecutedsuccessfully” (7) en la zona de mensajes y la respuesta del sistema en la parte de “Resultados” (8).

De esta manera podremos ver que configuración de intercalación tiene.



Comprobamos la configuración de SQL Collate mediante consulta SQL

A continuación vamos a ver otra forma de hacer la consulta de la SQL Collation en la que se hace una consulta a todas las instancias.

Dentro de la **consola de SSMS** hacer clic en “Nueva query” (9).

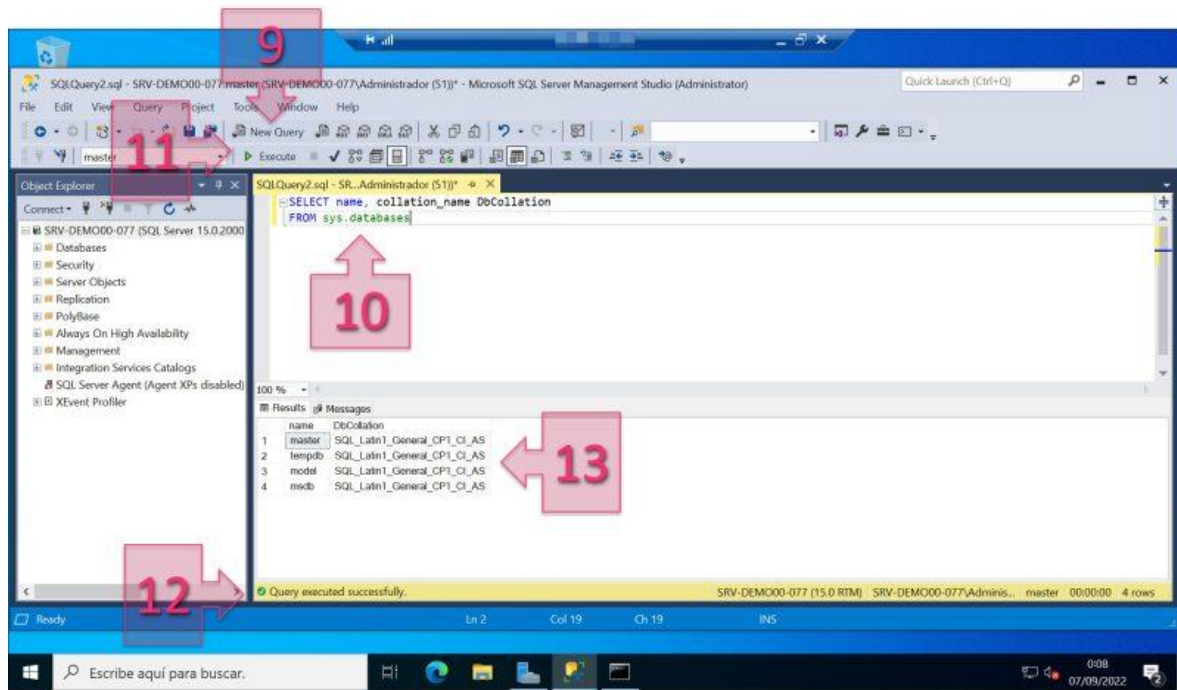
Dentro del espacio para consultas, se debe **introducir la siguiente consulta** (10):

```
SELECT name, collation_name DbCollation
FROM sys.database
```

Una vez hecho esto, se debe **hacer clic en “Execute”** (11).

Al igual que pasaba en el caso anterior, si todo va bien se podrá ver el **mensaje “Queryexecutedsuccessfully”** (12) en la zona de mensajes y la **respuesta del sistema en la parte de “Resultados”** (13).

De esta manera podremos ver que configuración de intercalación tiene cada una de ellas.



Consultamos la SQL Collation de todas las bases de datos configuradas

Con esto podemos dar por concluida la parte de consultas y podemos ponernos con la parte que interesa.

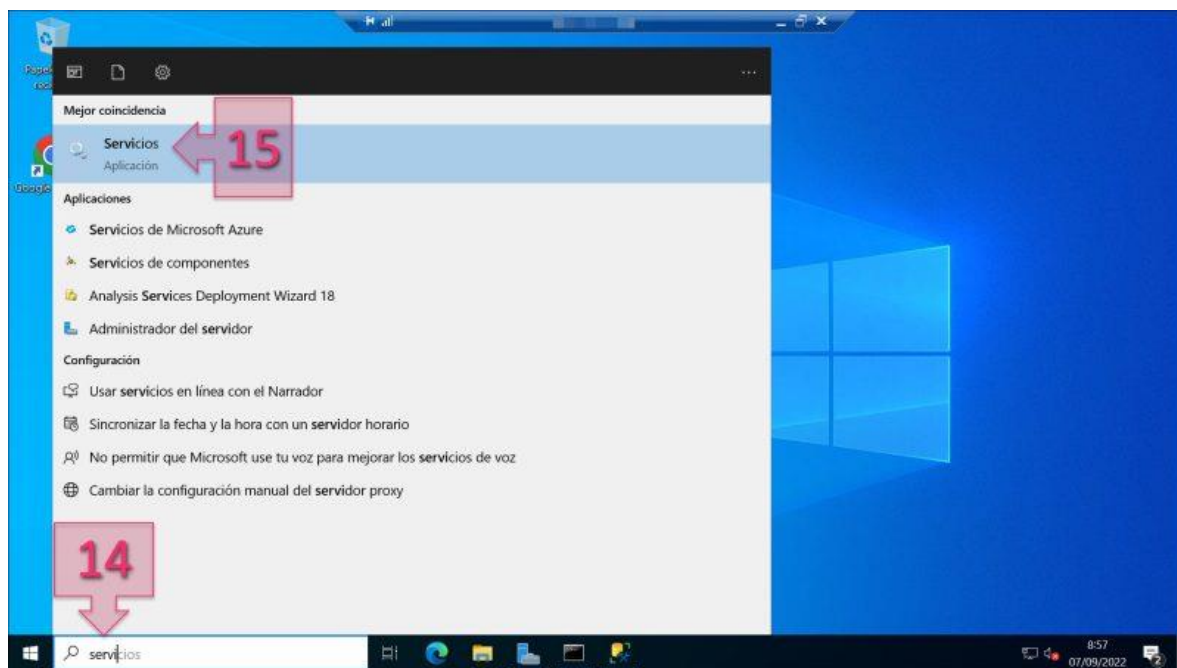
Paso 2. Configurar SQL Collate desde línea de comandos

Existen distintas maneras de cambiar el SQL Collate de una base de datos, pero vamos a describir el que quizá sea más sencillo de implementar, ya que lo que buscamos es minimizar el costo operacional pero manteniendo la eficiencia.

NOTA: *Es más que recomendable hacer copia de seguridad de todas las bases de datos que se hayan implementado en el servidor para evitar posibles pérdidas de información en caso de desastre.*

Para realizar esta operativa, lo primero que tendremos que hacer es **detener el servicio de SQL Server**. Esto permitirá hacer la operativa sin que haya conexiones de usuarios a las bases de datos y así evitar problemas de diversa índole.

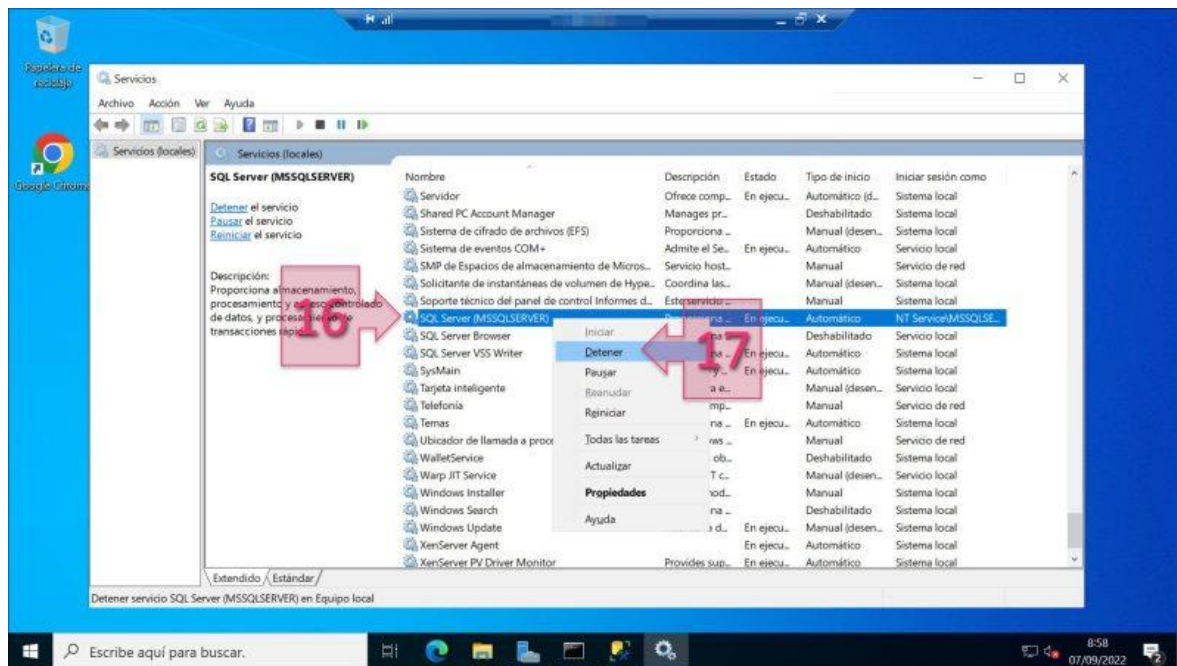
Para ello se debe **acceder a la consola de Servicios del sistema**, que puede ser accedida de muchas maneras, pero en este caso se debe **ir a la barra de búsqueda y escribir “Servicios”** (14), haciendo **clic sobre la aplicación que se muestra** (15).



Lanzamos la consola de administración de servicios del servidor

Dentro de la consola de Servicios se debe **buscar el servicio “SQL Server (MSSQLSERVER)”** que es el servicio que ejecuta la base de datos de Microsoft SQL Server y que vamos a detener para evitar problemas durante la operativa.

Una vez localizado el **servicio “SQL Server (MSSQLSERVER)”** se debe hacer **clic con el botón derecho** (16) y hacer **clic en “Detener”** (17) entre las opciones que se muestran. Esto **desencadenará la parada del servicio**, que no debería tomar mas que unos pocos segundos, aunque el tiempo que tome en esta acción dependerá de la carga del servidor.



Detenemos el servicio de SQL Server para poder hacer el cambio del valor de SQL Collate

Cuando estamos seguros de que el servicio está parado, podemos **lanzar una consola de CMD (Command) con privilegios de administrador**, que en realidad deberían ser heredados si estamos en una sesión con estos privilegios.

Desde la consola de CMD se debe **ejecutar el siguiente comando (18)**:

```
sqlservr -m -T4022 -T3659 -s»MSSQLSERVER» -
q»SQL_Latin1_General_CP1_CI_AS»
```

Donde tenemos los siguientes parámetros:

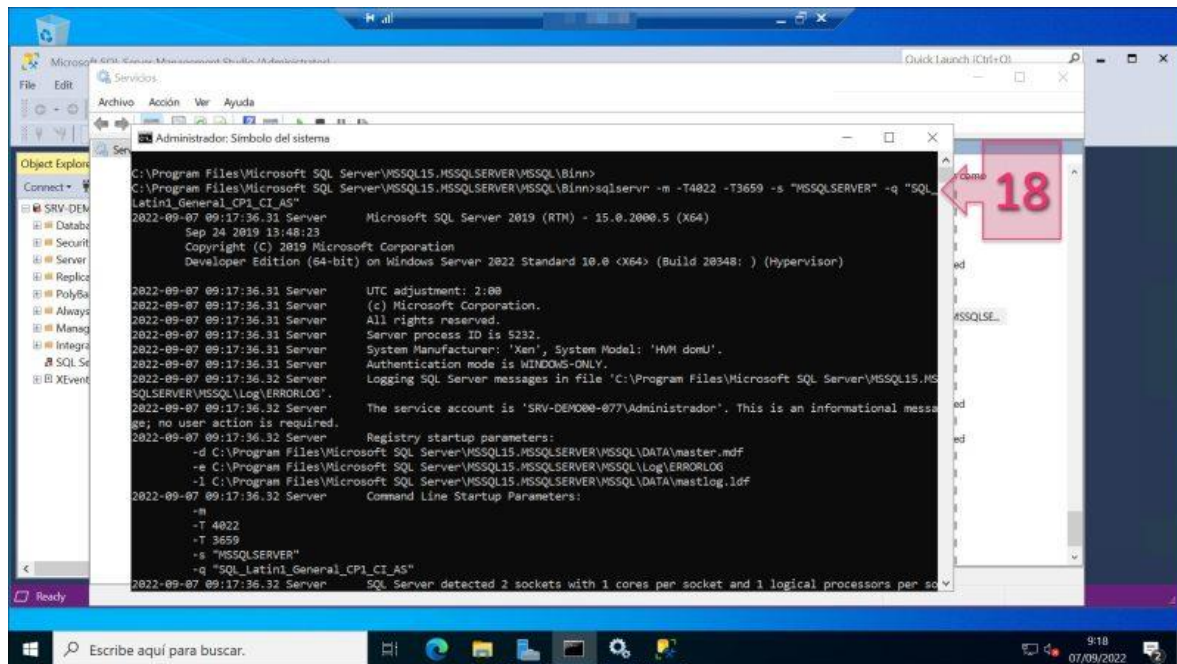
- -m: Hace que la ejecución se haga en modo mono usuario de administrador (singleuseradminmode).
- -T: Activa la traza durante la ejecución y el arranque.
- -s: Se usa para pasar el nombre de la instancia. Si se desconoce se puede usar la instancia por defecto de la base de datos, que es «MSSQLSERVER».
- -q: Para pasar la nueva intercalación (SQL Collate) que queremos aplicar, que en nuestro caso es «SQL_Latin1_General_CP1_CI_AS».

Además usamos dos flags de traza en esta ejecución:

- T4022: Hace un bypass del proceso de arranque de SQL Server.
- T3569: Deriva todos los errores de validación al “errorlog” durante el proceso de arranque.

Lo mismo podríamos hacer para pasarla a Francés, por ejemplo:

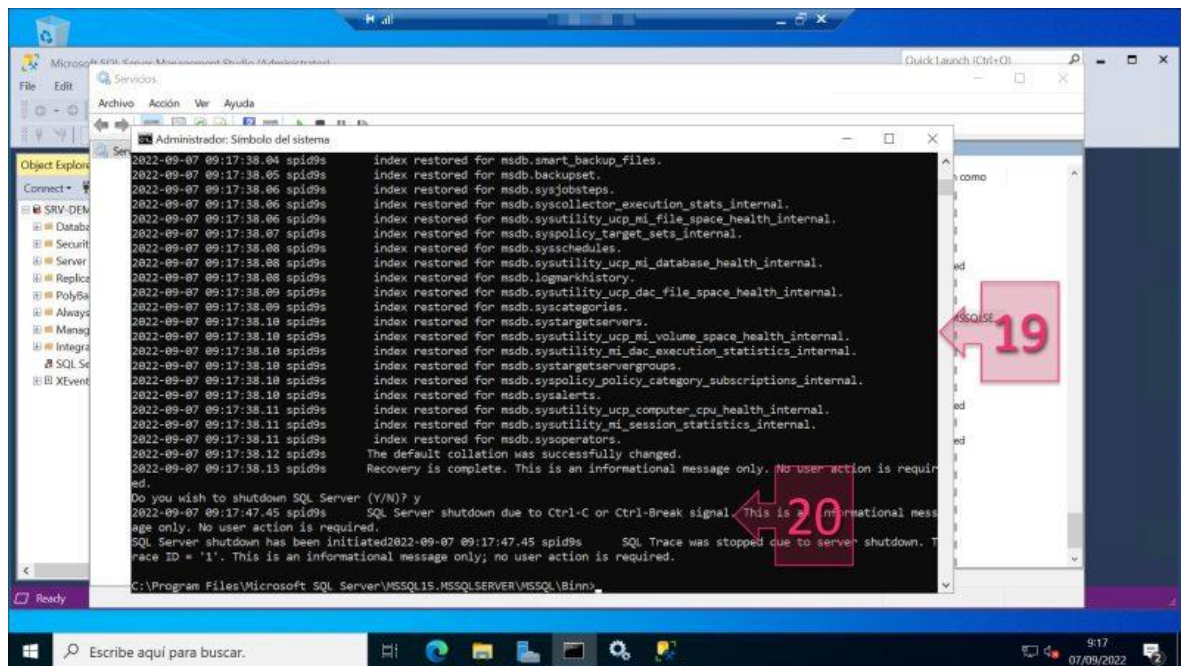
`sqlservr -m -T4022 -T3659 -s»MSSQLSERVER» -q»French_CI_AS»`



Lanzamos el comando para cambiar el SQL Collate de la base de datos

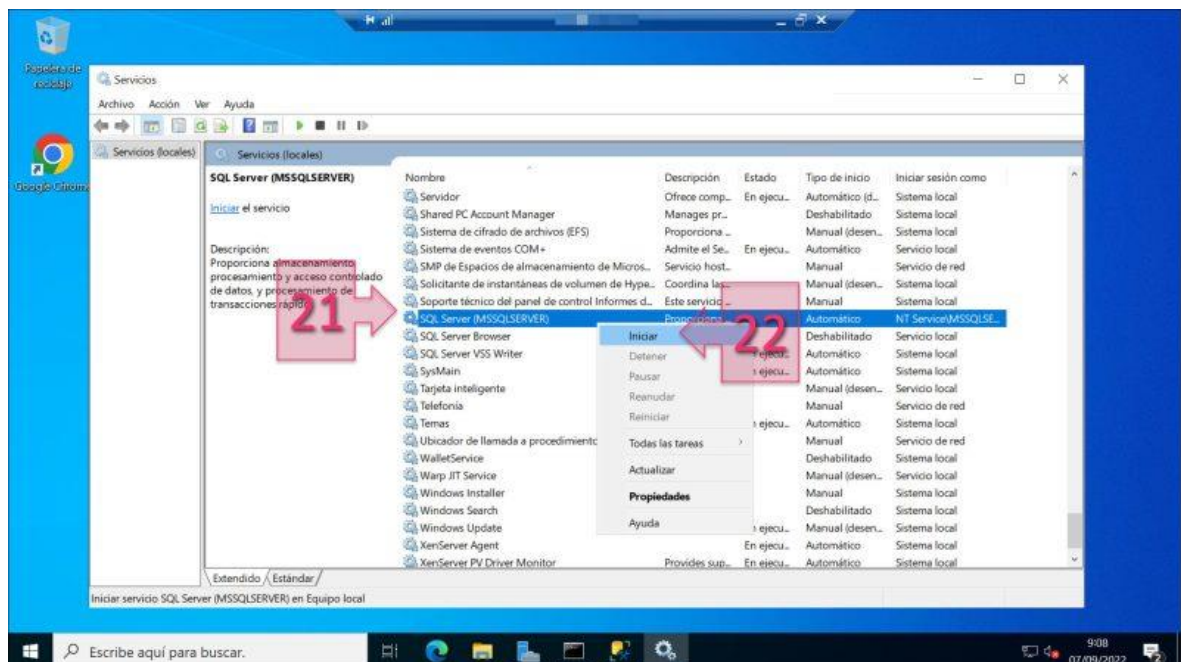
Durante la ejecución, se genera un **gran volumen de mensajes derivados de los cambios** (19) y que se muestran por pantalla, por lo que el administrado podrá revisarlos si así lo desea.

Al finalizar la ejecución, queda en stand by, esperando que el usuario le **mande una señal de fin de ejecución con un “Control + C”** (20).



Salimos de la ejecución del cambio de SQL Collate de manera manual

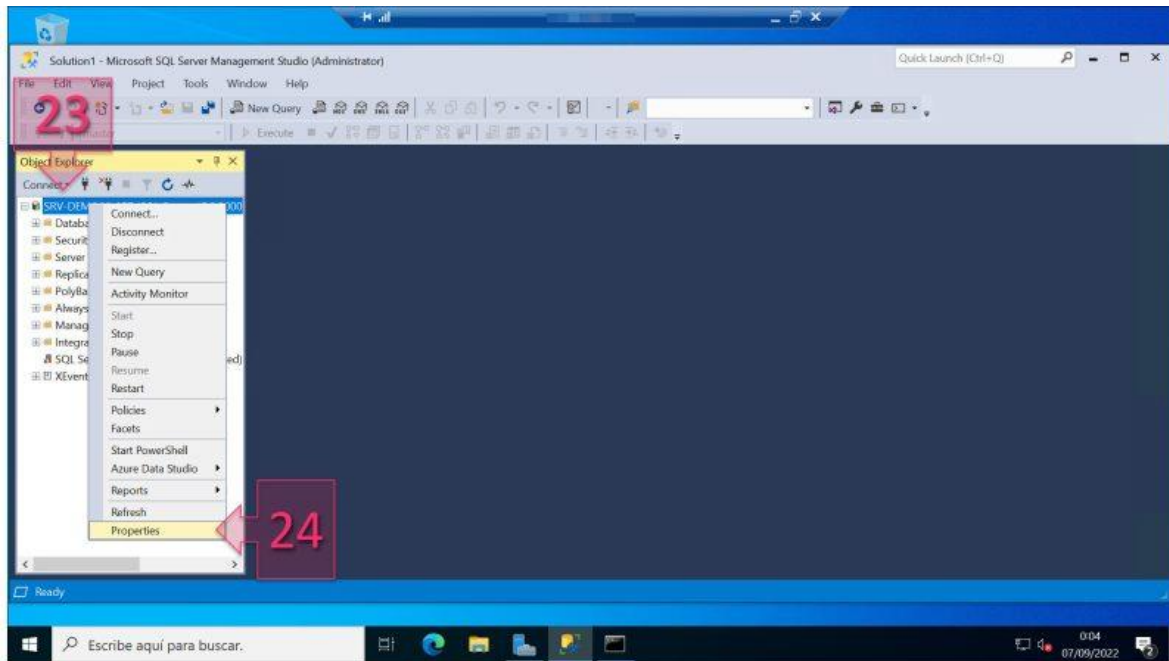
A continuación se debe volver al administrador de servicio y buscar el **servicio “SQL Server (MSSQLSERVER)”** se debe hacer clic con el botón derecho (21) y hacer clic en **“Iniciar”** (22) entre las opciones que se muestran. Esto desencadenará el inicio del servicio, que tardará unos pocos segundos, aunque nuevamente, el tiempo que tome en esta acción dependerá de la carga del servidor.



Iniciamos de nuevo el proceso de SQL Server

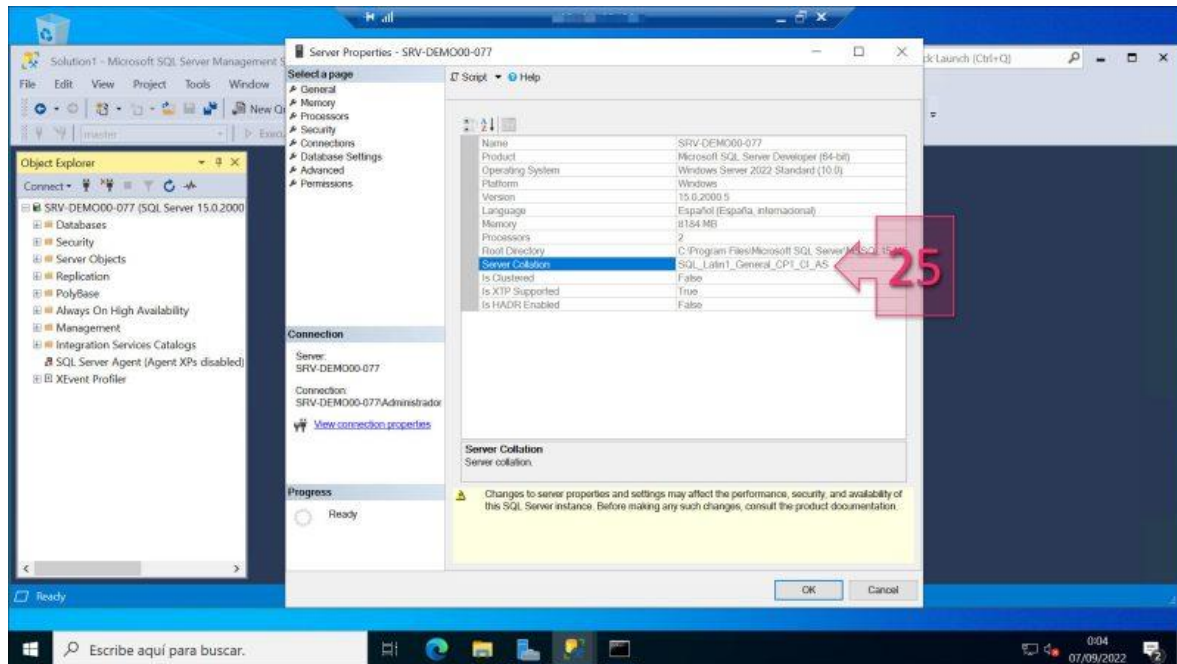
Tras esto, podremos comprobar de nuevo el estado de la intercalación lanzando nuevamente una consulta.

Para ello volvemos a **la consola de SSMS** y dentro del Explorador de Objetos, hacemos clic con el **botón derecho** sobre el nombre del servidor (23) y clic en **“Propiedades”** (24) entre las opciones que se muestran.



Volvemos a mostrar las propiedades de la base de datos

Una vez hecho esto, se muestra la ventana de las propiedades del servidor de SQL, entre las que se debe **comprobar el valor de “Server Collation”** (25) que debería mostrar la intercalación correcta.



Comprobamos que la configuración de SQL Collate se ha realizado con éxito

Y con esto tendríamos toda la operativa realizada y el valor de SQL Collate (SQL Collation) bien configurado.

Conclusiones

La intercalación, también conocida como **SQL Collation** o **SQL Collate**, es la **propiedad de la base de datos** que especifica los patrones de bits que van a **representar cada caracter del juego de datos** que utiliza nuestro idioma, así como la forma en que se podrán comparar u ordenar datos. Siendo un **aspecto crucial** para el buen funcionamiento de nuestras aplicaciones y bases de datos.

El no tener correctamente configurada esta propiedad **puede producir comportamientos indeseados**, y esta es la razón por la que se describen en este artículo varias formas para configurarlo, de manera que no de problemas para los usuarios que trabajan con sistemas ES_ES.