

Administración de Sistemas Xestores de Bases de Datos

Proyecto de evaluación

Beade Boán, Manuel



BBM_ASPACE

01 Preparación para trabajo

02 Modelado y Administración
de base de datos

03 Sistema de control de versiones

2021

ÍNDICE

ÍNDICE.....	1
INTRODUCCIÓN	1
PREPARACIÓN PARA TRABAJO.....	7
WINDOWS.....	7
<i>Creación de Máquinas Virtuales e instalación de Sistemas Operativos.....</i>	7
Creación de Máquina Virtual con Windows 10 [cliente]	12
Instalación de SQL SERVER 2017 en Windows 10 [Cliente]	14
Creación de Máquina Virtual con Windows Server2016 [controlador de dominio].....	21
Instalación de SQL Server 2017.....	27
<i>Configurar Firewall de Windows para permitir el acceso a SQL Server</i>	34
Configuración del Firewall en Windows 10	34
Creación de regla Inbound para permitir el puerto 1433	34
Creación de regla Inbound/Outbound para permitir el puerto 1434	38
Creación de regla Inbound/Outbound para permitir la red.....	43
Configuración del Firewall en Windows Sever 16.....	45
Creación de regla Inbound/Outbound para permitir el puerto 1434	45
Creación de regla Inbound/Outbound para permitir la red.....	49
<i>Instalación de SSMS.....</i>	51
Problemas en el inicio de SSMS.....	53
<i>Configuración de la red en VMWare.....</i>	54
Configuración del adaptador de red en Windows Server 2016.....	54
Configuración del adaptador de red en Windows 10	58
Configuración del Dominio	62
Instalar el rol Servicios de dominio de Active Directory	72
Unir la máquina con Windows 10 al Dominio.....	78
Acceder desde SQL Server a otras instancias de otros equipos del dominio.....	89
<i>Explicar Instalación Desatendida</i>	92
LINUX.....	143
<i>Creación de MV e instalación de Ubuntu 18.04.....</i>	143
Configuración del firewall de Ubuntu.....	145
Configuración de la red de Ubuntu.....	146
Unir Ubuntu al dominio	149
<i>Instalar SQL Server 2017 / 2019.....</i>	161
<i>Instalar GUI: Azure Data Studio</i>	165
<i>Instalación y uso de Sqlcmd</i>	167
MODELADO BASE DE DATOS.....	175
MODELADO CON SQL DATA MODELER	175
<i>Instalación de Oracle</i>	175
<i>SQL Developer - Oracle Express</i>	178
<i>Conexiones.....</i>	180
Desde CMD	180
Desde GUI	181
<i>Desbloquear Usuario HR</i>	187
<i>Bases de Datos de Ejemplo</i>	204

<i>Modelado Lógico - Relacional - Fisico Proyecto</i>	217
Subvistas.....	219
Modelo Lógico	222
Modelo Relacional	223
Generar Archivo DDL	224
<i>SQL Data Modeler a SQL Server</i>	251
Convertir el modelo Lógico-Relacional a SQL Server	255
INSTALAR BASES DE DATOS DE EJEMPLO EN SSMS	257
<i>Pubs desde Script</i>	257
<i>Northwind con ATTACH</i>	259
<i>Adventureworks2017 desde BACKUP</i>	261
<i>WideWorldImporters con BACPAC</i>	263
PRUEBAS PREVIAS	266
FILEGROUP	281
FILESTREAM	283
FILETABLE	291
BASES DE DATOS CONTENIDAS	298
PARTICIONES	298
TABLAS TEMPORALES	317
TRIGGERS	326
TABLAS IN MEMORY	349
BACKUP ALL CURSORS	359
STORED PROCEDURE BACKUP	362
SISTEMA DE CONTROL DE VERSIONES	366
GIT COMO SISTEMA DE CONTROL DE VERSIONES DISTRIBUIDO	366
SENTENCIAS GIT	367
<i>Sentencias GIT REPOSITORY</i>	367
<i>Sentencias INDEX / STAGING AREA</i>	368
INSTALANDO GIT	370
<i>Instalación del "cmd" de GIT</i>	370
<i>Entornos gráficos</i>	374
PRIMEROS PASOS CON GIT	374
<i>Crear un repositorio en GitHub</i>	374
<i>Uso del Command Prompt de GIT</i>	375
Inicio con GIT Bash.....	375
Creación del archivo README.md	382
Ignorar archivos con .gitignore	383
Reset vs Revert	384
GIT RESET	384
GIT REVERT	388
Git Branch.....	395
ENTORNO GRÁFICO EN WINDOWS: GIT TORTOISE	403
BIBLIOGRAFÍA	408

INTRODUCCIÓN

El concepto de este trabajo se basa en tres grandes bloques: La Preparación para el trabajo, el Modelado de base de datos y por último GIT.

En cuanto a la primera parte, en la Preparación para el trabajo presentaré tres máquinas virtuales:

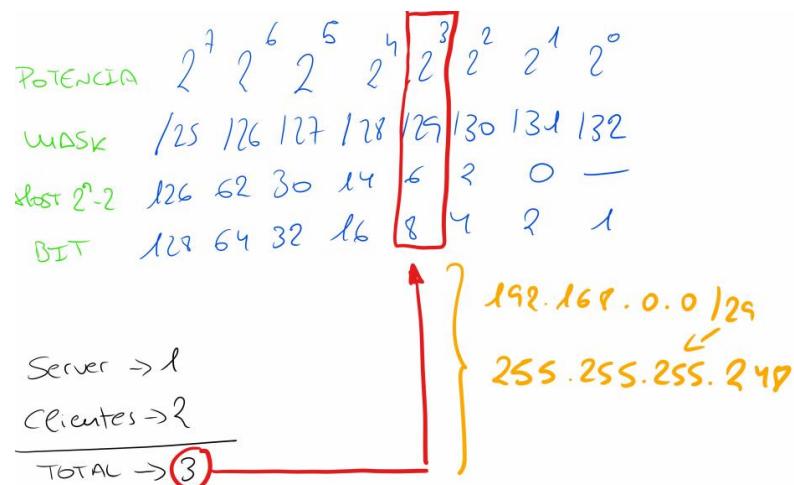
- Una máquina Windows Server 16
- Una máquina cliente Windows 10
- Una máquina cliente Linux Ubuntu 18.04



¿Qué va a pasar con estas máquinas? Lo que va a suceder es que se va a explicar cómo se crean esas máquinas virtuales, y una vez hecho esto, interconectarlas entre las tres creando primero una red específica para ellas. Para envolver un poco las asignaturas que estamos cursando, he decidido hacer subnetting para ajustar lo máximo posible la red.

Tenemos tres equipos en un entorno de red privada, lo que voy a usar es una IP de clase C con una máscara correspondiente de clase C. Para esta clase, los tres primeros bits del primer octeto se reservan para ser los bits iniciales que identifican a la clase. Por ello, el rango de direcciones son 192.168.0.0 – 192.168.255.255 y la máscara por defecto 255.255.255.0.

A partir de esto hice el siguiente cálculo:



Con esos cálculos se obtiene la siguiente tabla, en la cual se especifican las IP de cada equipo:

EQUIPO	IP	MÁSCARA	Gateway	IP DNS
Server 16	192.168.0.2	255.255.255.248	En caso de haberla, sería 192.168.0.1	192.168.0.2
Windows 10	192.168.0.3	255.255.255.248		192.168.0.2
Ubuntu 18.04	192.168.0.4	255.255.255.248		192.168.0.2
Gateway	192.168.0.1	255.255.255.248		
Broadcast	192.168.0.7	255.255.255.248		
Sobrante	192.168.0.5	255.255.255.248		
Sobrante	192.168.0.6	255.255.255.248		

NOTA: Lo que está en rojo es que no hace falta y sobra. En caso de inclusión de un router y otros dos clientes, entrarían en esas IP.

Lo ideal hubiera sido crear un servidor de DHCP y que sirviese las IP justas para los equipos y excluir el resto para evitar algún fallo de seguridad. En las pruebas efectuadas esto daba muchos problemas y decidí meter el direccionamiento de manera estática.

Un problema añadido de las máquinas es el **Firewall**.



En este proyecto dejó marcadas tres reglas que permiten el acceso de los equipos entre sí con los puertos **1433** y **1434** de **SQL Server** y la red **192.168.0.0/29**.

Con todo esto hecho, pasamos a configurar el dominio, uniendo las tres máquinas virtuales entre sí en un **Active Directory**.



Después de todo este compendio se instalan los programas en la máquina cliente (yo lo hice antes y tuve que reinstalarlo todo).

Los que instalamos son los citados a continuación:

Windows Server 16

SQL Server 2017

SSMS

Linux

SQL Server 2017

Azure data Studio

SQLCMD

Windows 10

Visual Studio Code

SQL Server 2017

SSMS

SQL Developer

Data Modeler

Git Bash

Git Tortoise



Para un supuesto de que haya que instalar nuevas máquinas, haré una explicación de una instalación desatentida por si se desease instalar un equipo nuevo (de los cuales ya tengo reservadas las IP).

Para finalizar, probaré la conexión entre las máquinas, intentando conectarme desde la máquina **cliente Linux** y el **server Windows** a la **máquina cliente Windows 10**.

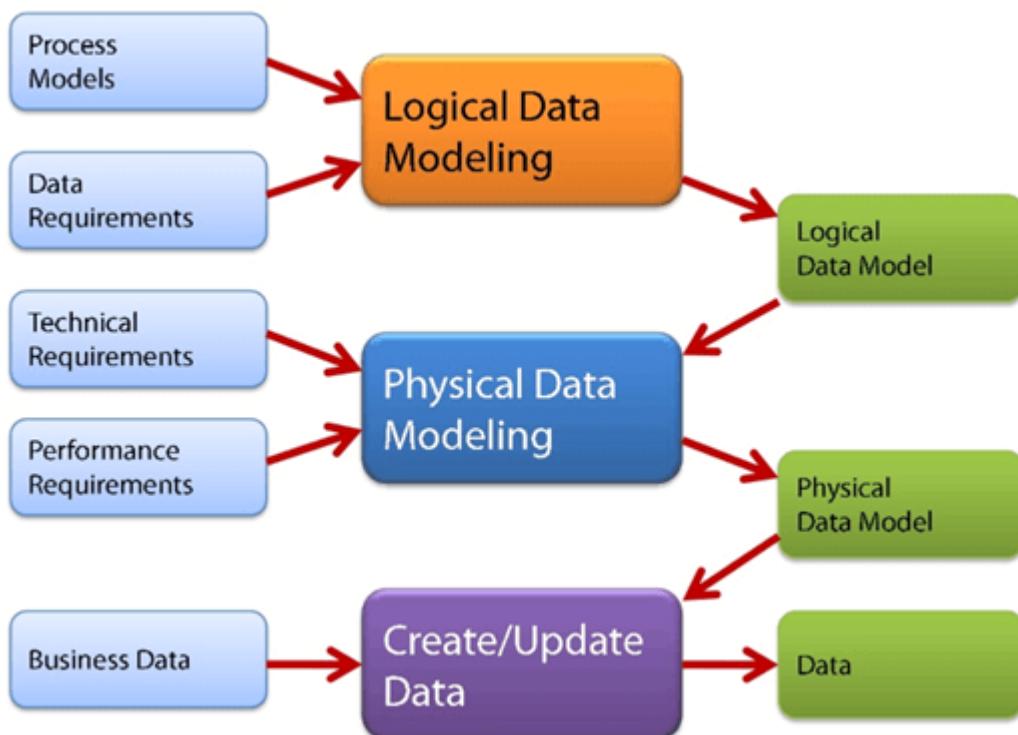
En la parte de **Modelado de base de datos**, explicaré las instalaciones de algunos de los programas que no expliqué en el anterior apartado. Empezaré por meter el **SQL Developer**, uno de los programas de Oracle que no metimos en el proyecto del año pasado.



Después de explicar su instalación y unos pasos de uso como puede ser conectarse a él o **desbloquear el usuario HR**, pasará a explicar el **Data Modeler**.

Ahí ya me meteré de lleno en el **modelado de datos**, primero con unas bases de ejemplo, y luego con **mi proyecto**.

Para desempeñar el proyecto, seguí un poco este esquema:

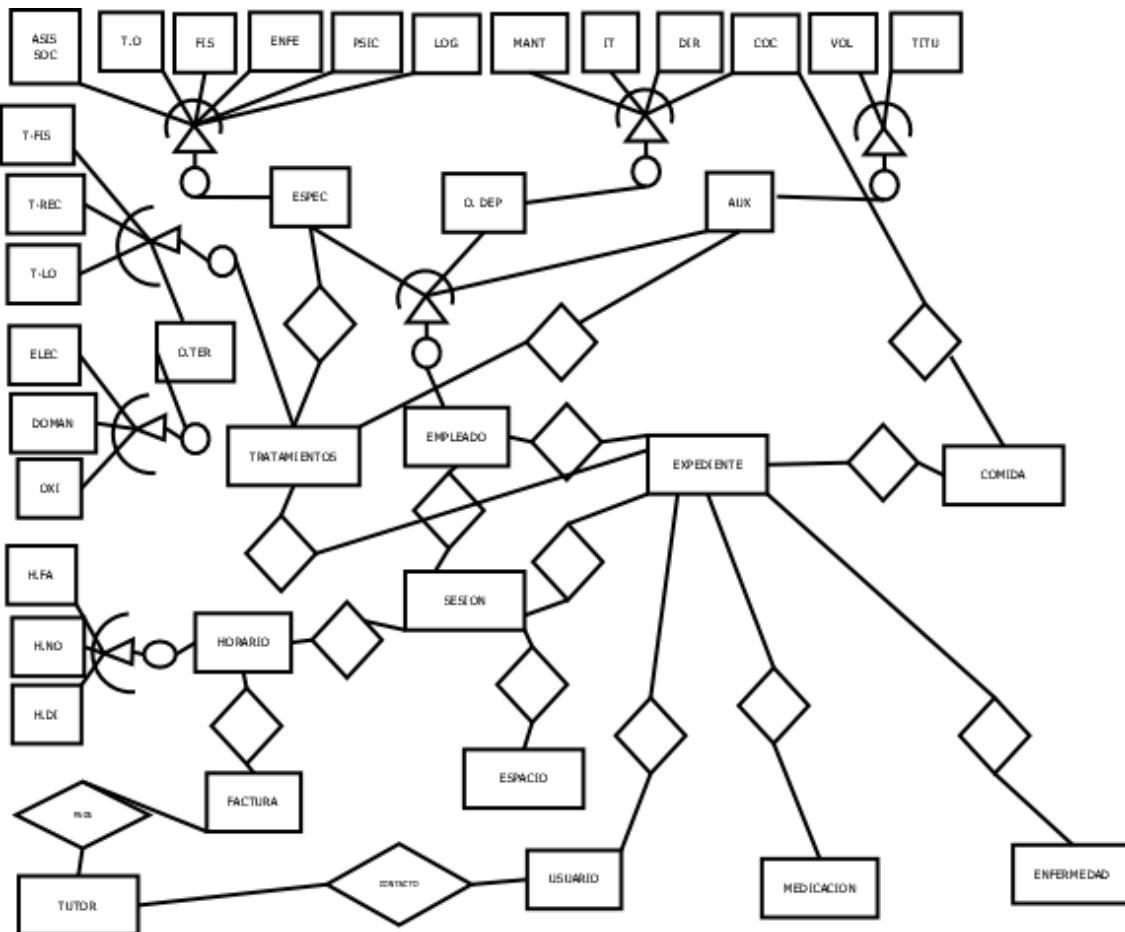


Basado en esto, mi proyecto **BBM_ASPACE** es una base de datos que nace de la necesidad de que organizaciones como ASPACE que se trata de un lugar para tratar pacientes que sufren sobre todo de parálisis cerebral, tengan un mejor control en una base de datos y no en archivos de Excel como están en la gran mayoría de casos. Por ello, me he dedicado a recolectar datos sobre lugares como este y he preguntado a varias personas que han trabajado allí. Desde una **fisioterapeuta** que trabajó allí hasta un **informático** que trató de poner un poquito de orden en aquel lugar.

Las tablas en las que se centra este proyecto son cuatro:

- Usuarios → Las personas con parálisis que son tratadas
- Expediente → Un registro de todo lo que se hace con ellos
- Empleados → Todo el personal que trabaja allí
- Tratamientos → Todos los métodos que se emplean en los usuarios

En base a estos bloques se desarrolla toda la actividad, que quedaría en un primer boceto de la siguiente manera:



En base a esto, desarrollo el diseño lógico – relacional en el programa. Una vez hecho, hago la ingeniería para obtener el archivo .DDL que meteré en el SSMS, pasando así mi diseño en **Oracle a SQL Server**. Con la base de datos creada hago el Diagrama de mi proyecto, y para terminar hago unas pruebas con el SSMS, que será instalar las bases de datos de ejemplo desde diversos lugares:

- Pubs desde script
- Northwind con ATTACH
- Adventureworks2017 desde BACKUP
- WideWorldImporters con BACPAC

Con esto hecho, paso a desarrollar la base de datos de mi proyecto, desde hacer una **inserción de valores**, pasando por **triggers**, hasta los **procedimientos almacenados** para realizar **backups**.

Con todo esto acabado, toca subir el proyecto a la nube. Esto lo haré con el **Sistema de Control de Versiones**, en este caso, **GIT**.



Tras una explicación de su funcionamiento, las sentencias más comunes, instalaré su **CMD**, en este caso **Git Bash**, y desde ahí subir los archivos al repositorio <https://github.com/BBMASAX/BBMSPACE.git>. Para acabar, dejaré el entorno gráfico **Git Tortoise**.



Se adjunta el enlace del repositorio: <https://github.com/BBMASAX/BBMSPACE>

PREPARACIÓN PARA TRABAJO

WINDOWS

Creación de Máquinas Virtuales e instalación de Sistemas Operativos

Voy a crear una máquina Virtual utilizando **VMWARE** (VM es el acrónimo *Virtual Machine*). En este proyecto se van a utilizar dos softwares **VMWARE** y **VirtualBOX**. La versión que vamos a utilizar es Workstation (estación de trabajo), cuyo icono es el siguiente:

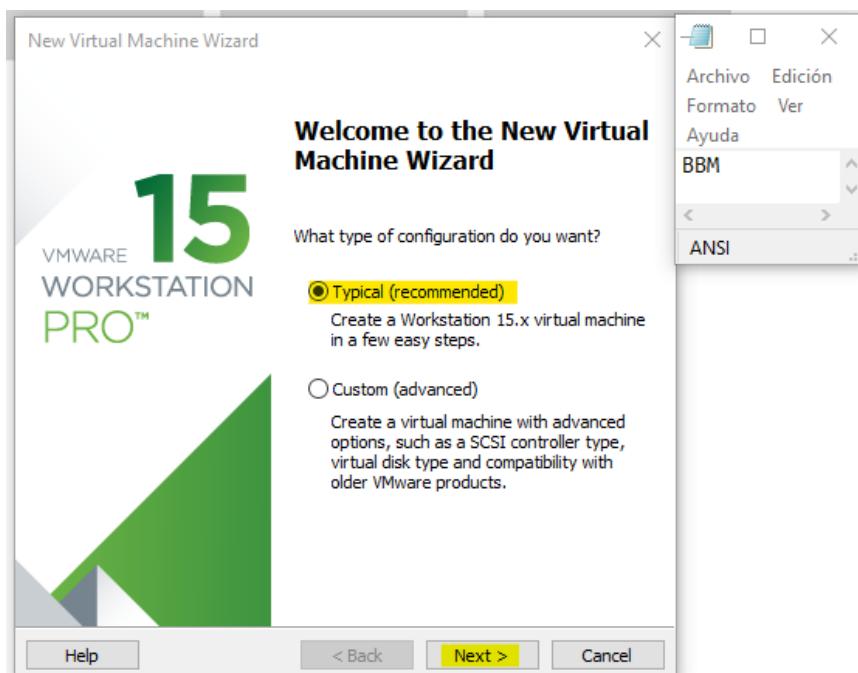


Lanzamos y comenzaríamos a crear una máquina virtual guest (invitado).

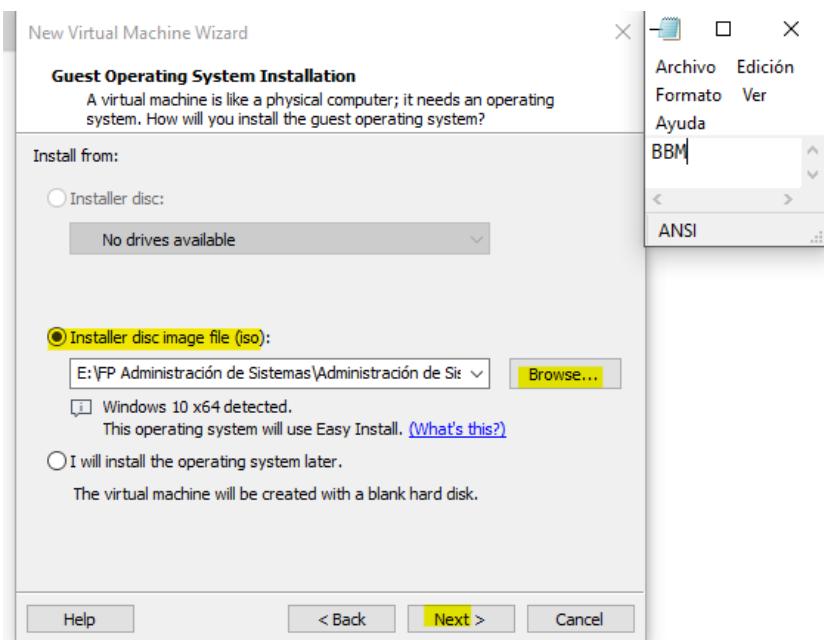
Hacemos clic en el botón **Create a New Virtual Machine** y comenzaríamos con la instalación:



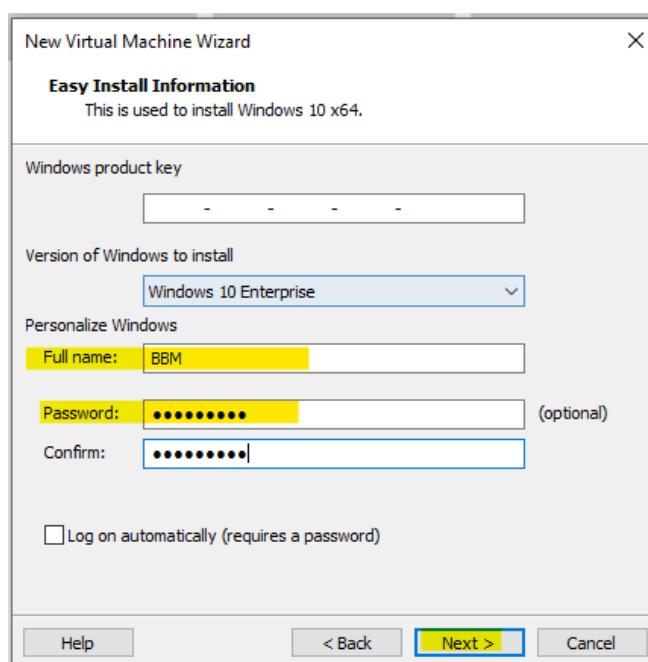
Me sale una primera ventana donde podemos elegir entre una configuración *Typical* (recomendada) o *Custom* (personalizada). Para este proyecto la instalación necesaria es la *Typical*.



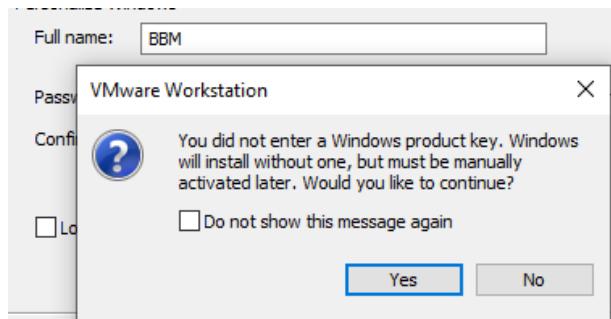
En la siguiente ventana tendremos que elegir el sistema operativo. Hay tres opciones: Instalarlo desde disco, desde una ISO o hacerlo más tarde. La decisión tomada será instalarlo ahora desde un archivo (ISO), al cual le tendremos que especificar la ubicación del archivo. Un detalle es que el Windows que estamos seleccionando es de 64 bit.



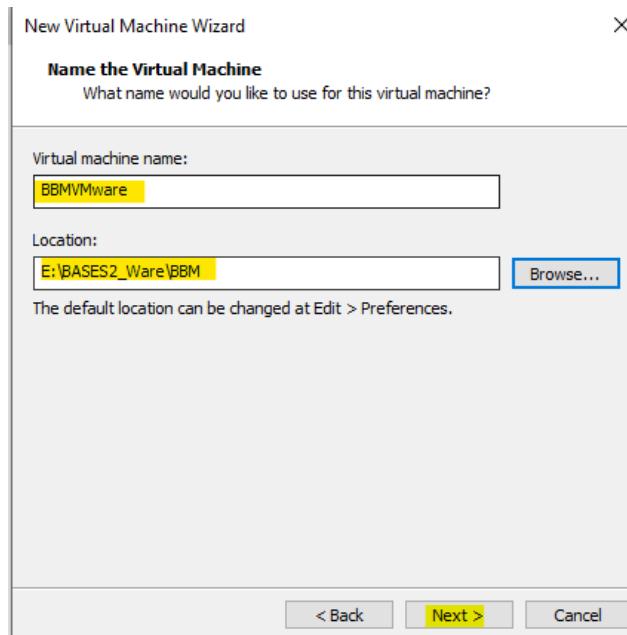
En la siguiente nos pide clave de producto, no dice la versión de Windows (Enterprise). En la parte de personalización de Windows pongo de nombre las iniciales de los apellidos y el nombre (**BBM** en mi caso), y como contraseña la más básica (*Abcd1234.*) para simular un entorno de trabajo:



NOTA: Como no tenemos una clave activada, nos va a mostrar un mensaje. Le damos a Yes.

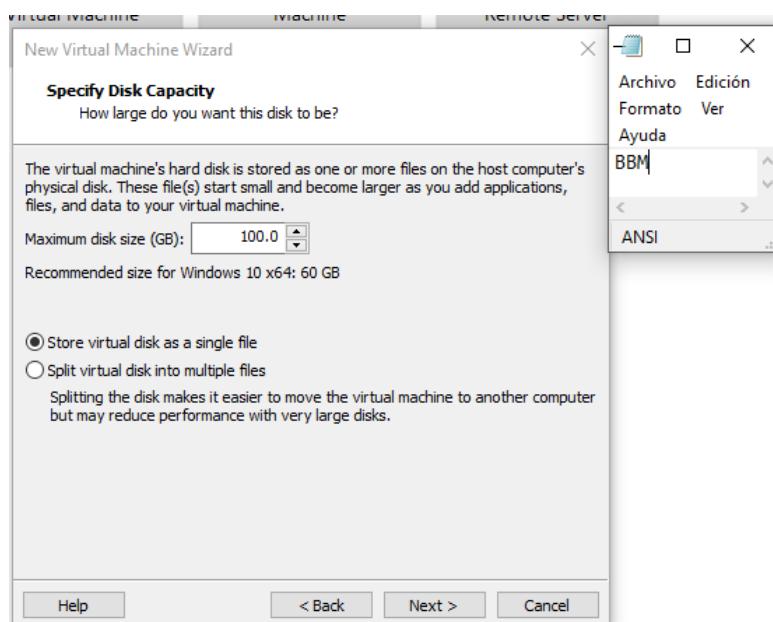


La siguiente ventana es para ponerle el nombre a nuestra máquina (en la cual intentaremos que todo tenga nuestro nombre) y una ubicación (en este caso el disco duro). Esta ubicación es una carpeta física:

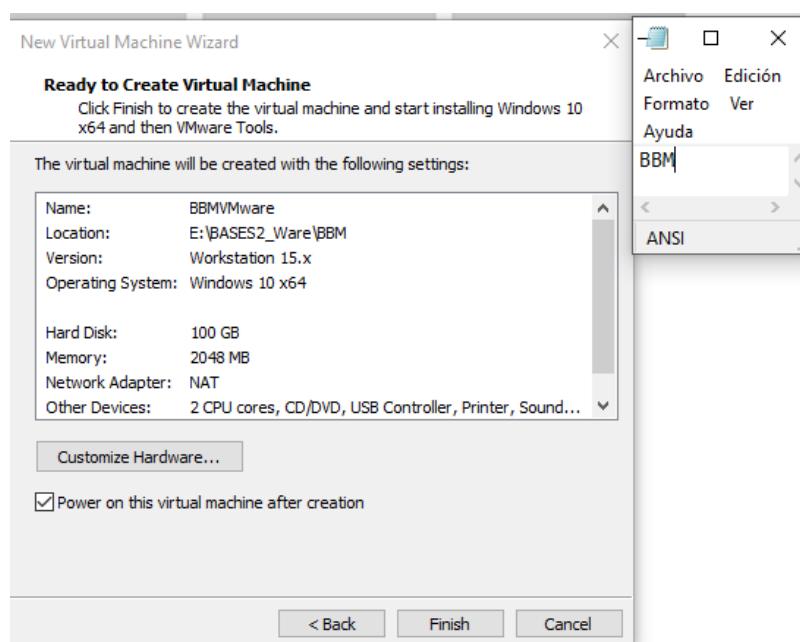


La siguiente ventana me muestra el tamaño del disco. Como el proyecto se realiza a lo largo del curso, le daremos 100GB al menos porque luego aumentar el disco suele dar muchos problemas. En realidad yo reservo 100GB, pero al principio no ocupa 100, y va creciendo a medida que lo vamos llenando.

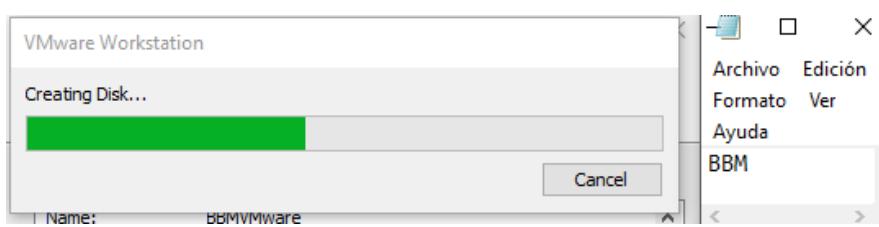
En las opciones, cambio a la opción **Store virtual disk as a single file**. Le doy a **Next:**



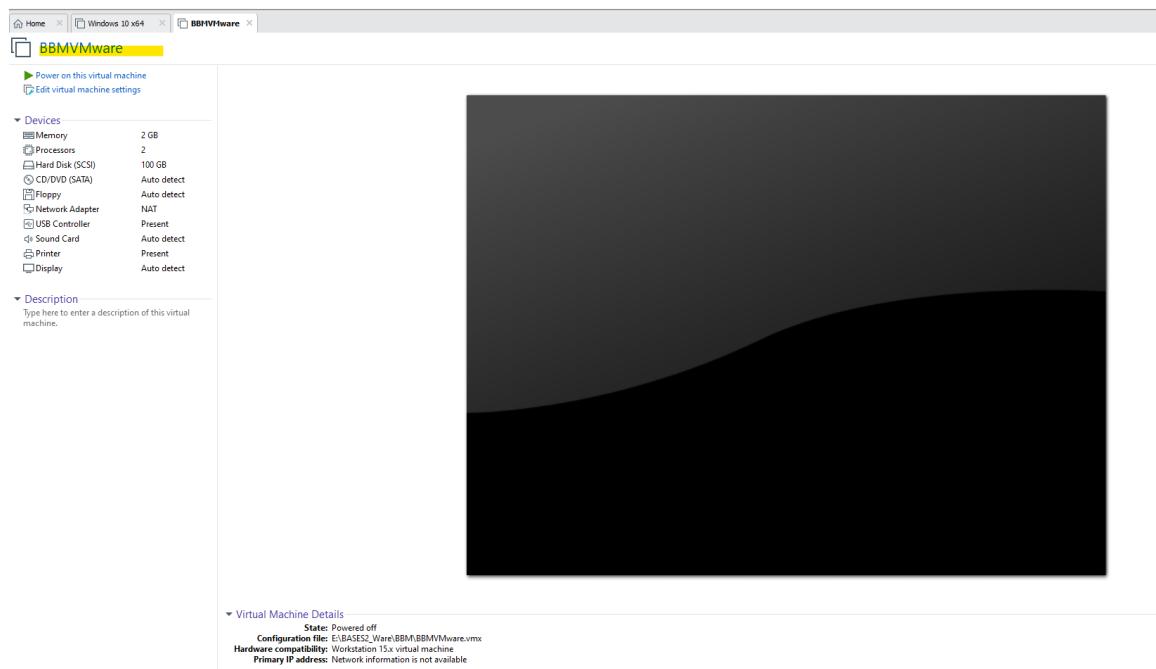
En la siguiente ventana, tendremos una especie de resumen y le damos a *Finish*.



Como se ve, está creando el disco:

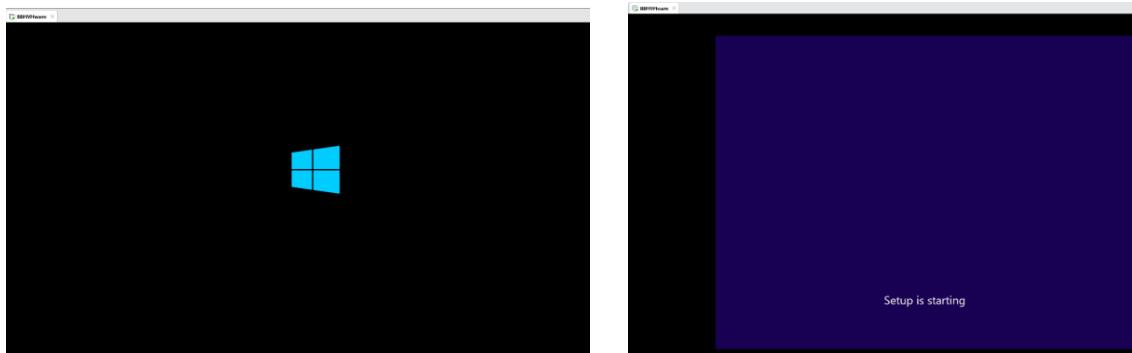


La presentación de la máquina creada se vería de esta manera:

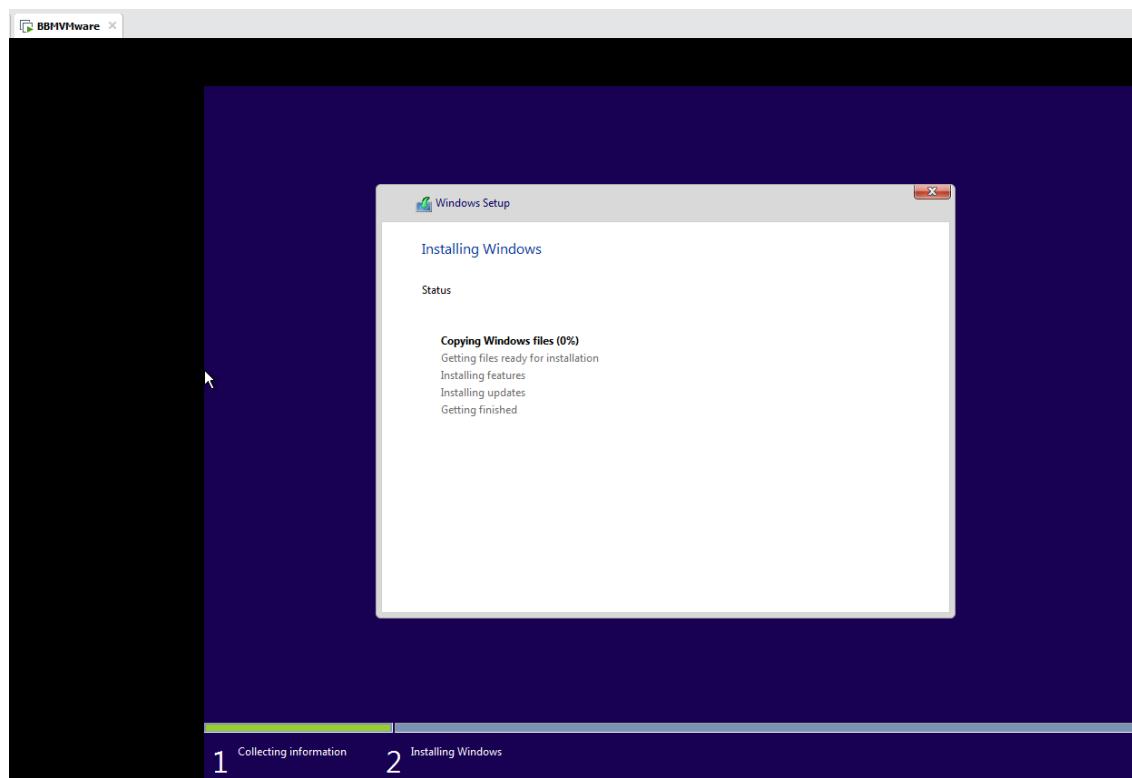


Creación de Máquina Virtual con Windows 10 [cliente]

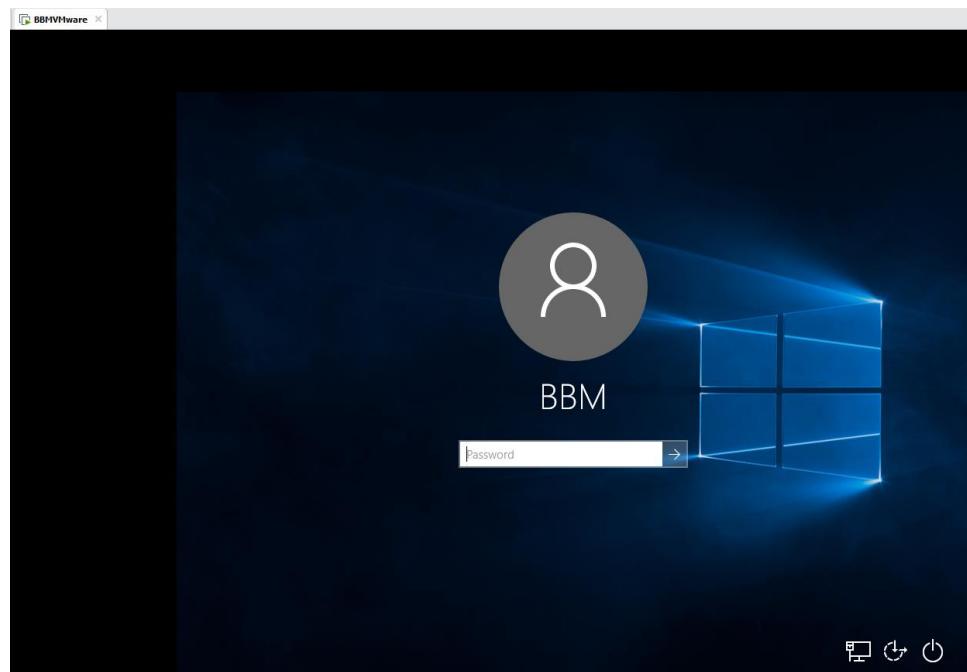
Comienza la carga de Windows 10:



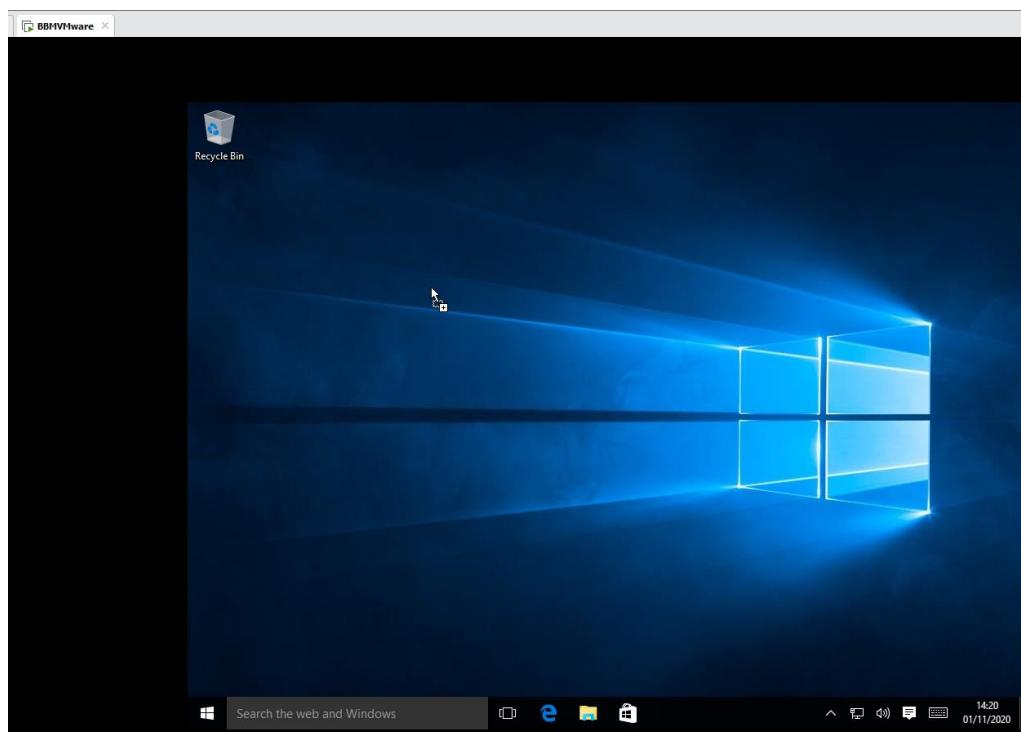
Empieza la instalación con recopilación de la información. Una vez acaba se reinicia



Salto pasos de carga y pongo la ventana para empezar con el usuario en la que meteremos la contraseña puesta antes (Abcd1234.)



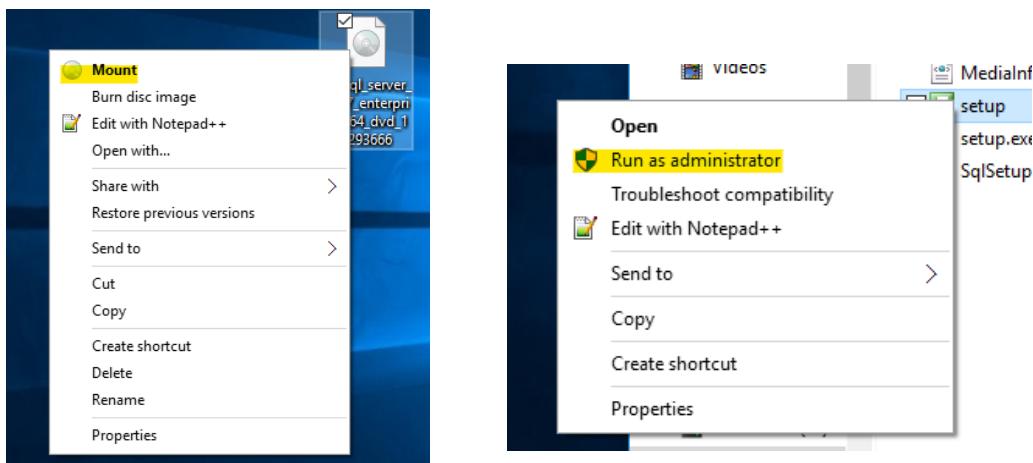
Lo único que hay que comprobar es si la máquina deja copiar archivos de fuera para empezar a trabajar



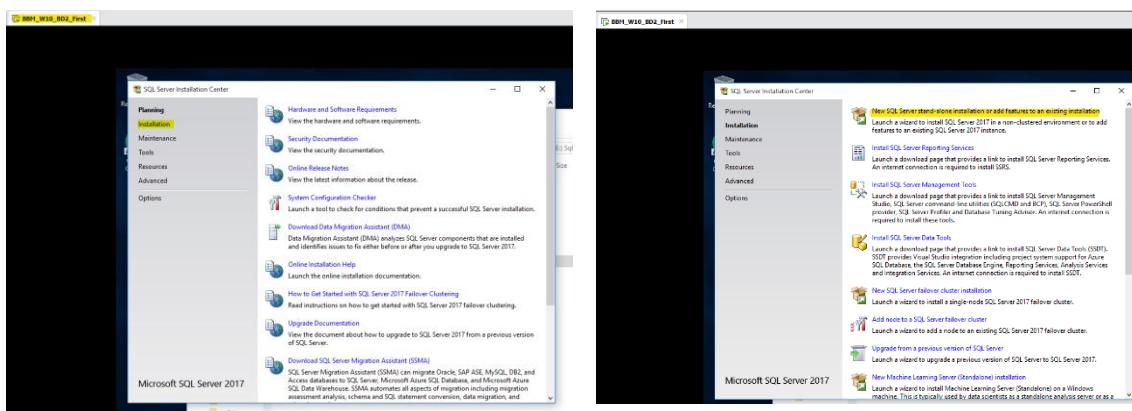
Instalación de SQL SERVER 2017 en Windows 10 [Cliente]

Lo primero que hacemos en la instalación es hacer botón derecho sobre la ISO y clicar en Mount. He de aclarar que lo voy a hacer con SQL Server 2017, porque al parecer la versión 2019 da bastantes problemas.

Nada más montar, iniciamos como administrador, clicando con el botón derecho sobre el ícono setup. Aceptamos la instalación.

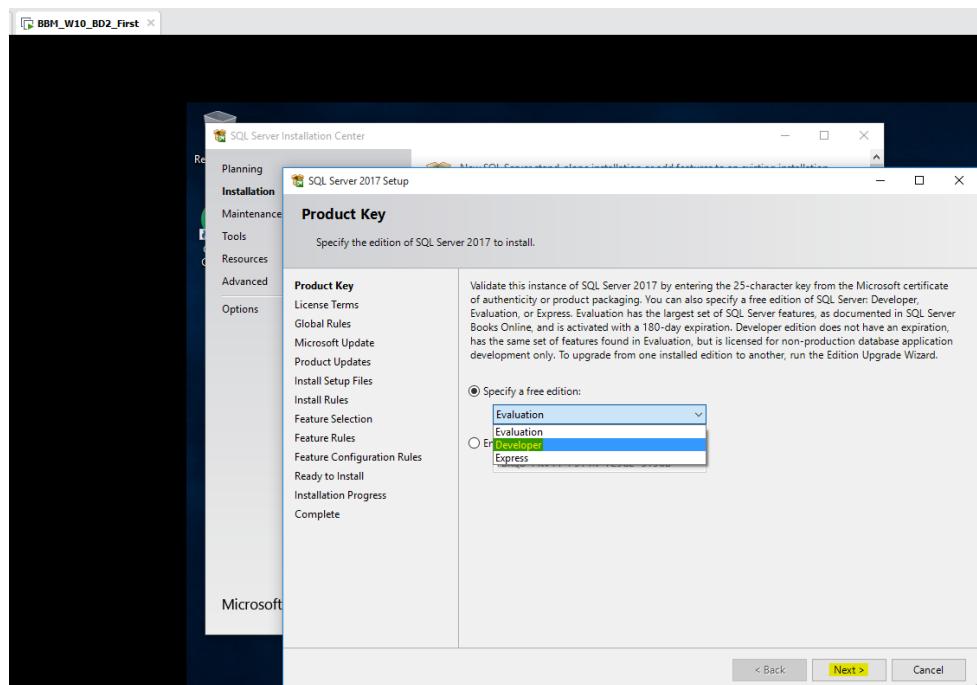


La siguiente pantalla que sale es un panel de ayuda. Son links que en ocasiones valen para conectar en SQL Server. En este caso, lo que se va a usar es Installation. Dentro de todas las opciones que nos encontramos, esta vez vamos a utilizar *New SQL Server stand-alone or add feature to an existing installation*. Podemos distinguir entre una instalación stand-alone (único equipo) o un clúster (agrupación). En este caso instalaremos un equipo (stand-alone) y le añadiremos algunas características.

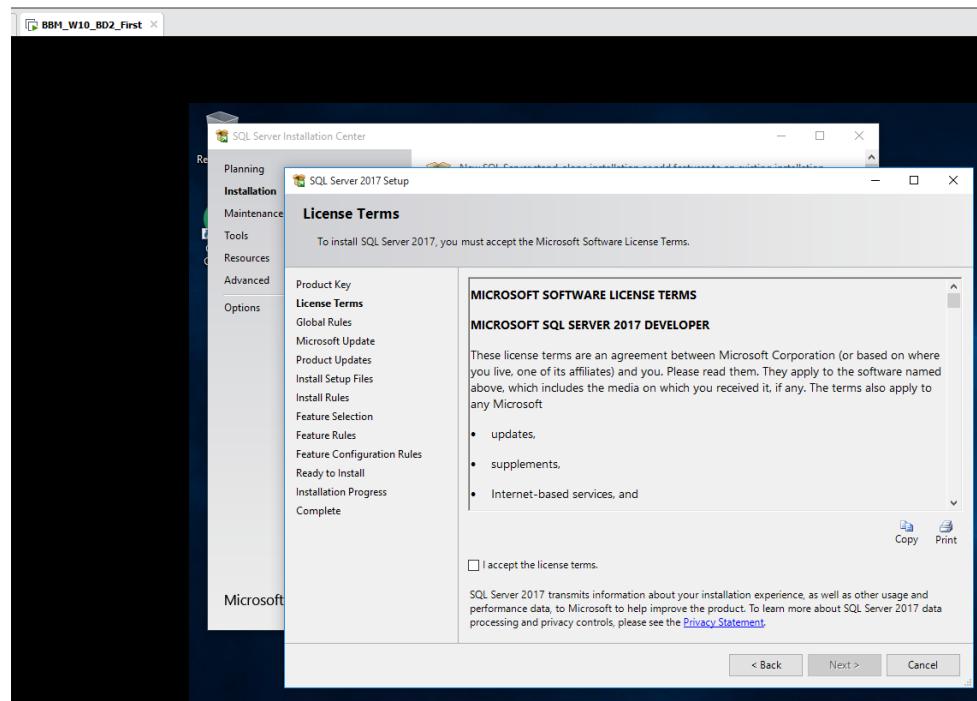


Se abre una nueva ventana donde me dice que especifique un tipo de edición. *Evaluation* sería la opción trial; *Express* es la opción gratuita que solemos utilizar en pequeñas empresas, y *Developer* es la que tiene todas las características sin

ponerlas en producción. En caso de haber pagado la licencia, elegiríamos la opción de poner una key. En este caso elegimos la opción *Developer*.

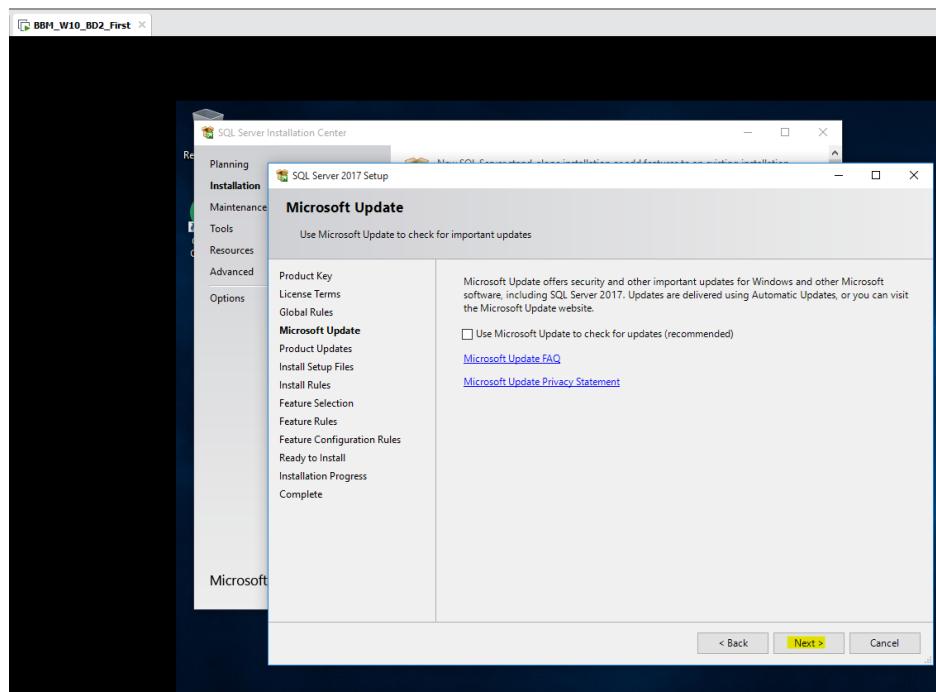


Aceptamos los términos de la licencia y le damos a continuar.

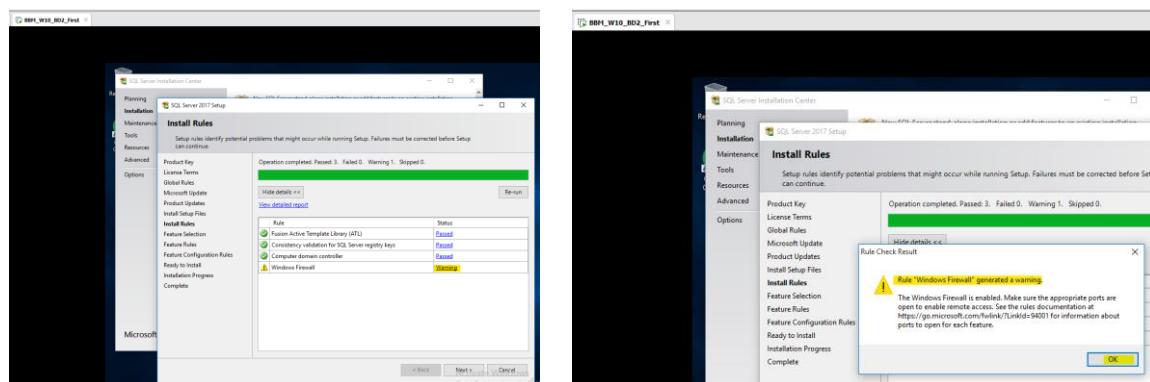


La siguiente ventana hace como una previa de lo que tenemos instalado en nuestro equipo, y nos pregunta si queremos realmente que busque actualizaciones

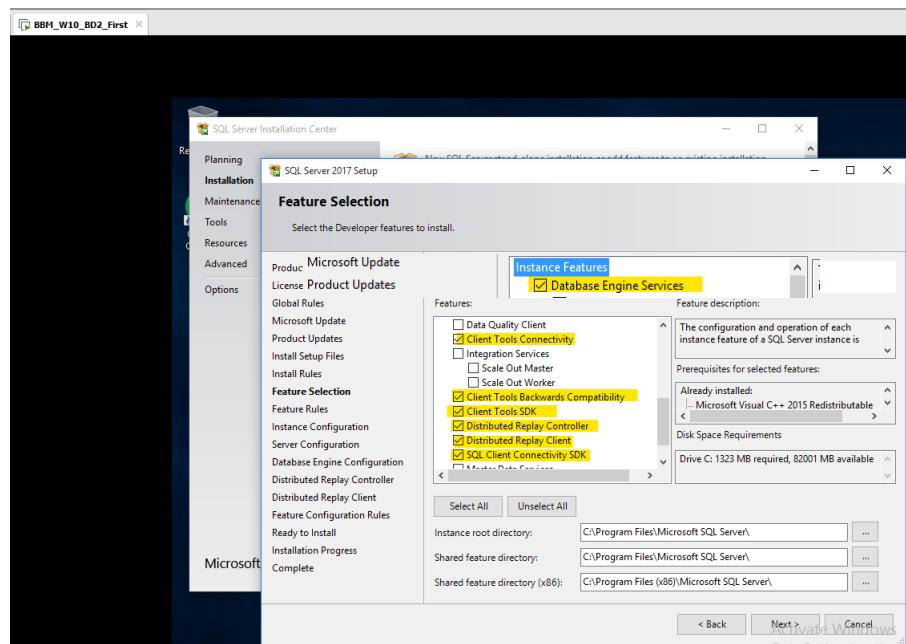
automáticamente. En este caso no nos interesa tal cosa, así que le damos a continuar sin clicar en el botón de *check*.



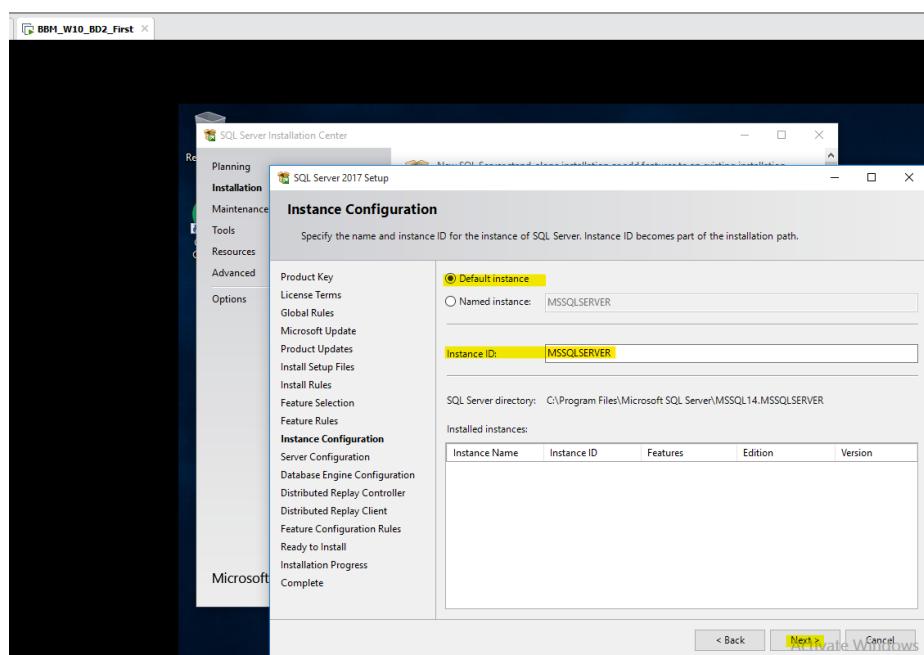
La siguiente es una ventana donde empezará con la instalación. Los check verdes quieren decir que la instalación se ha completado de manera satisfactoria, rojo un error, y amarillo, un *warning* que nos avisa de que hay algo que no va a funcionar correctamente hasta que se solvete (en este caso el firewall). En la parte de *Status*, tenemos un link, que da una información adicional en una ventana nueva sobre el evento que seleccionemos. En este caso voy a utilizar el Warning para que quede reflejado. Tras leer la información que me interese, le daría a Next:



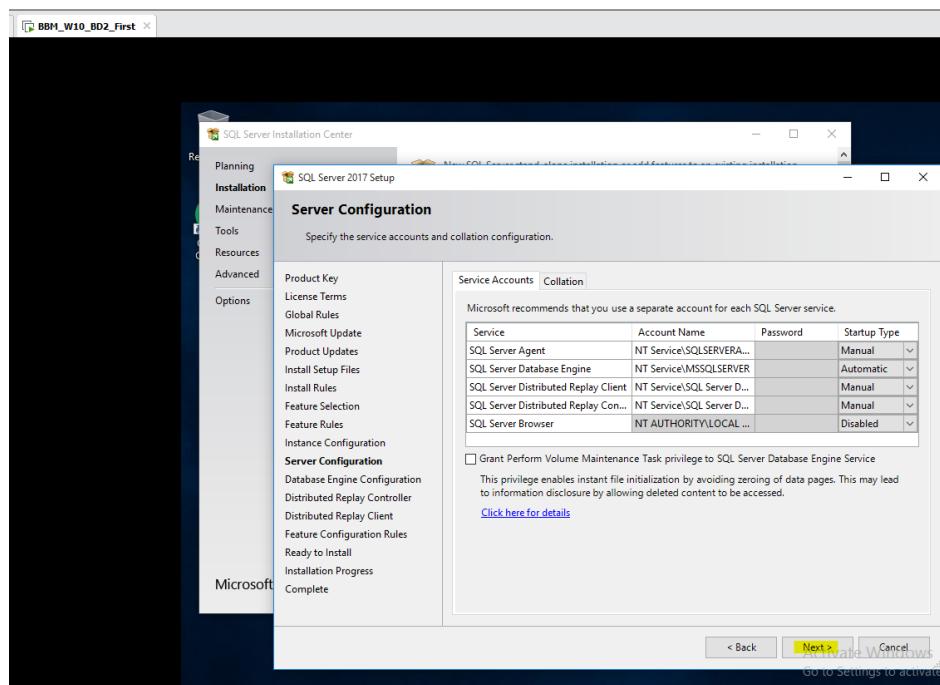
En la siguiente ventana, elegiremos qué elementos quiero elegir. Elegiremos los servicios de bases de datos (*Database Engine Services*). También elegiré en las características compartidas todas las que tengas que ver con cliente: *Client Tools Connectivity*, *Client Tools Backwards Compatibility*, *Client Tools SDK*, *Distributed Replay Controller*, *Distributed Replay Client*, *SQL Client Connectivity SDK*. Siempre se pueden cambiar estas características a posteriori. Todos los directorios los dejo por defecto.



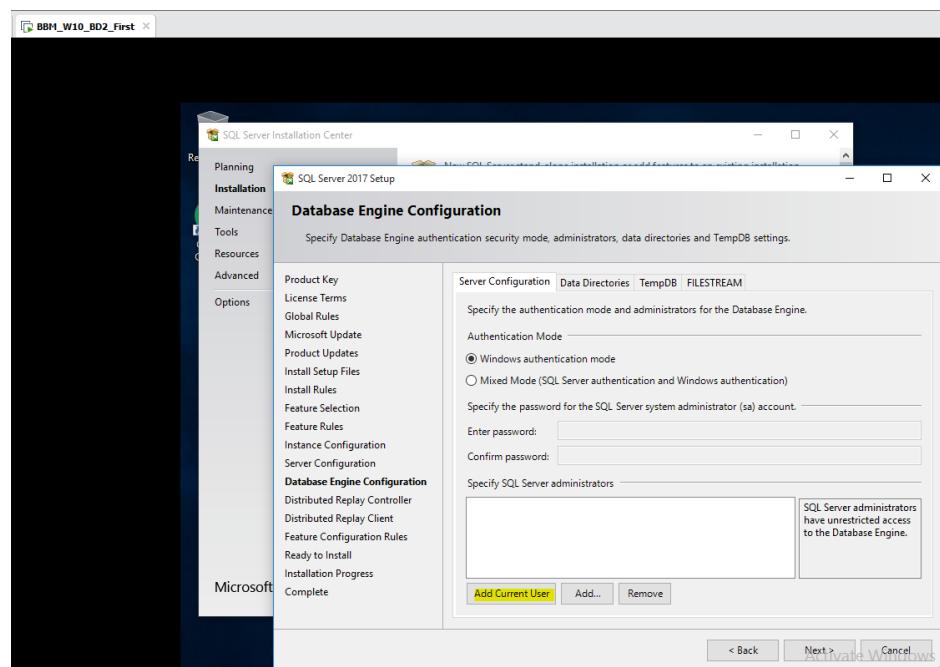
En la siguiente parte, configuraremos una instancia. Lo que aquí llama una instancia es un servidor de base de datos. La primera es la instancia por defecto. Si fuese la segunda, elegiría *Named instance*. Elijo *Default instance* y le doy a Next:



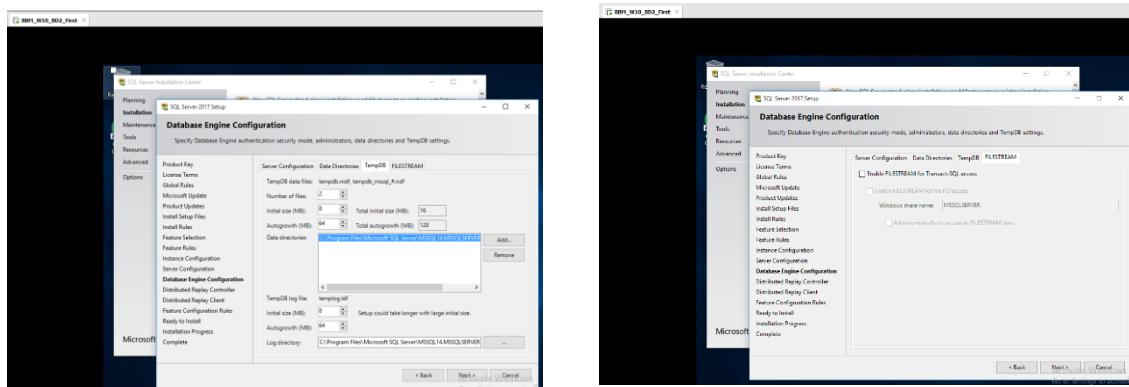
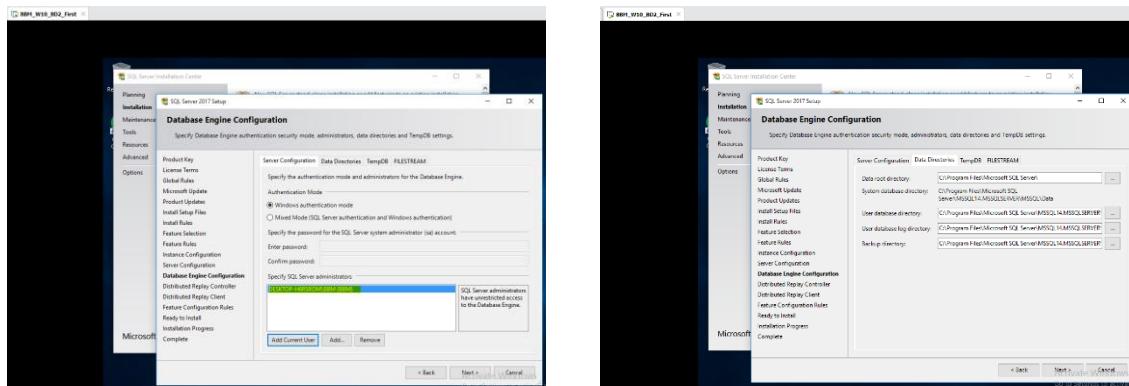
Al darle a siguiente nos van a aparecer los servicios. Los que aparecen son los que necesitamos. El más importante es el SQL Server Database Engine (el sistema de gestión). Todo se instalaría en cuentas de dominio, pero para nuestras pruebas queda por defecto:



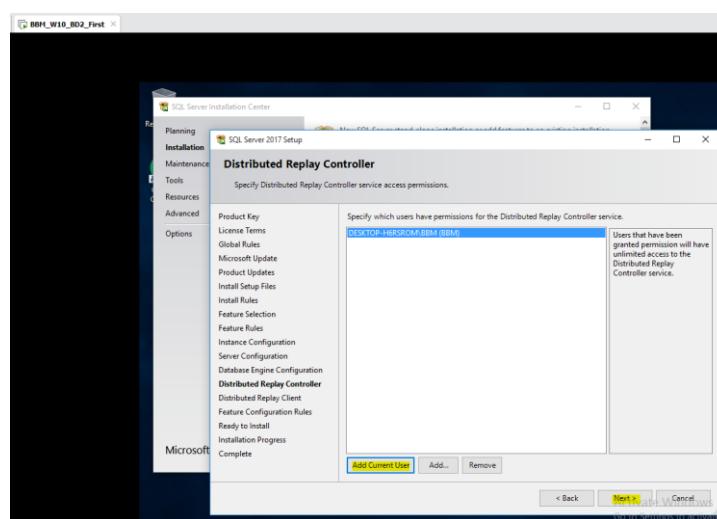
Lo siguiente que vamos a configurar es el modo de autenticación. En este caso dejamos la opción de Windows, y le vamos a dar a añadir usuario actual (*Add Current User*).



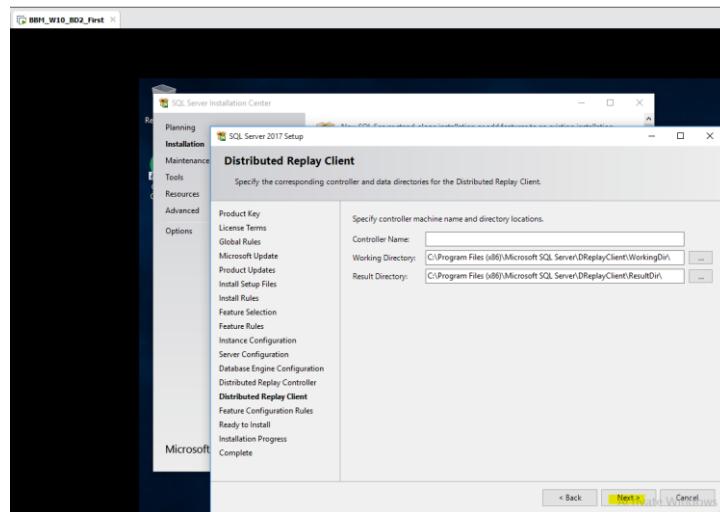
Vemos que al cabo de unos segundos aparece nuestra cuenta. Si exploramos el resto de las pestañas, vemos que podríamos variar los directorios, modificar la base TempDB, o subir archivos



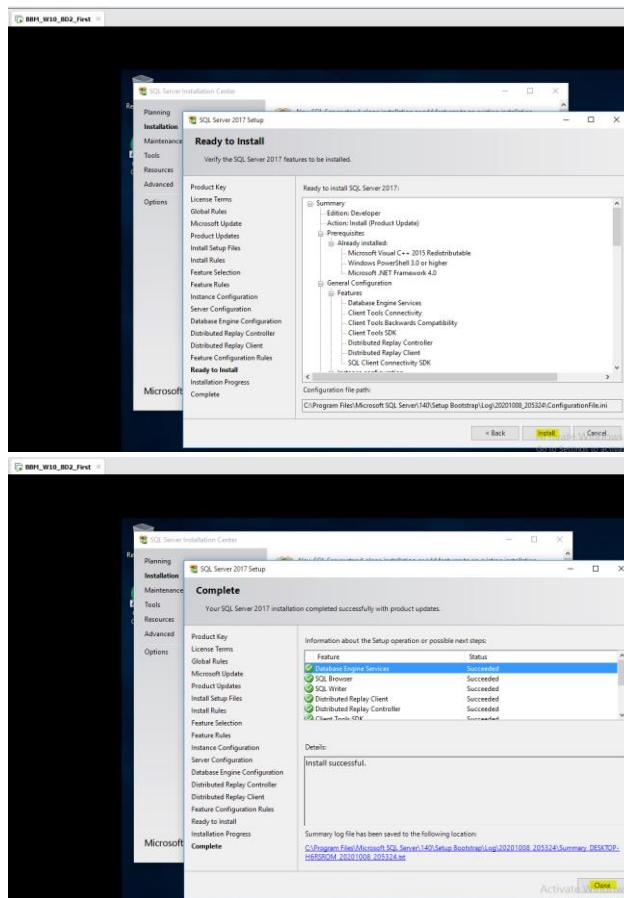
En la siguiente venta le especificamos el usuario con permiso dando a *Add Current User*:



La siguiente ventana le damos a Next



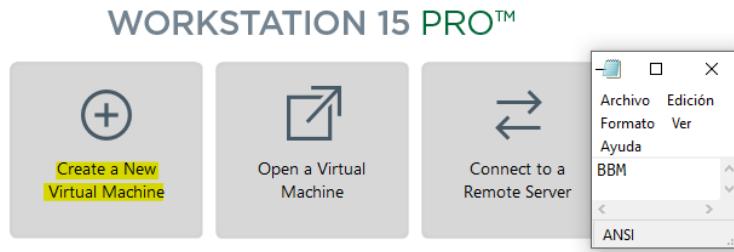
En esta ventana nos da un resumen antes de instalar. Comprobamos que todo está ok e instalamos SQL SERVER. Como apunte, el *Configuration Path* sirve para las instalaciones desatendidas, cosa que explicaré más adelante.



La última página es un resumen de todo lo instalado. Le damos a *close* y como en todos los programas reiniciamos el ordenador.

Creación de Máquina Virtual con Windows Server2016 [controlador de dominio]

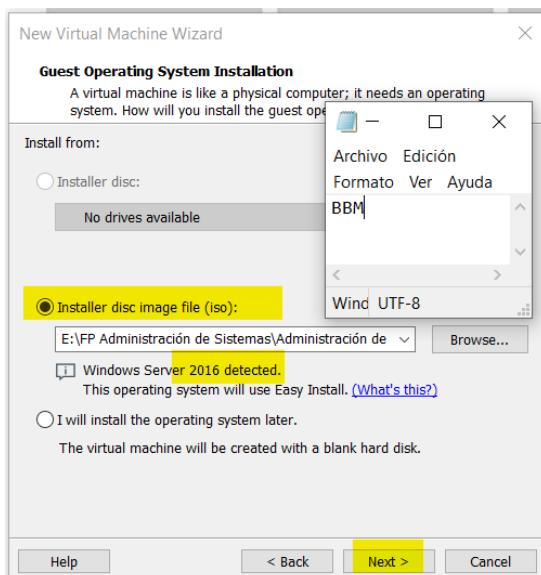
Al igual que en la creación de la [máquina de Windows](#), le damos al botón *Create a New Virtual Machine*:



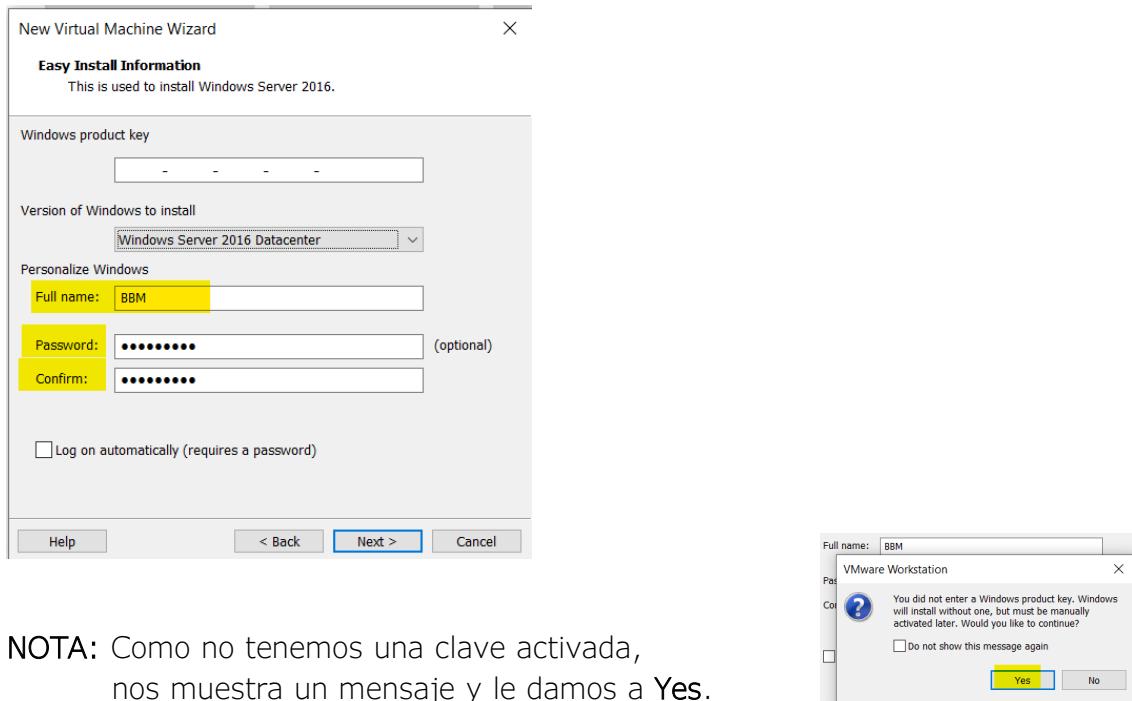
Cuando se abre la ventana es todo prácticamente igual. Elegimos la opción *Typical*, y hacemos clic en *Next*:



En la siguiente ventana, elegimos instalar desde una imagen de disco [*Installer disc image file (iso)*]:

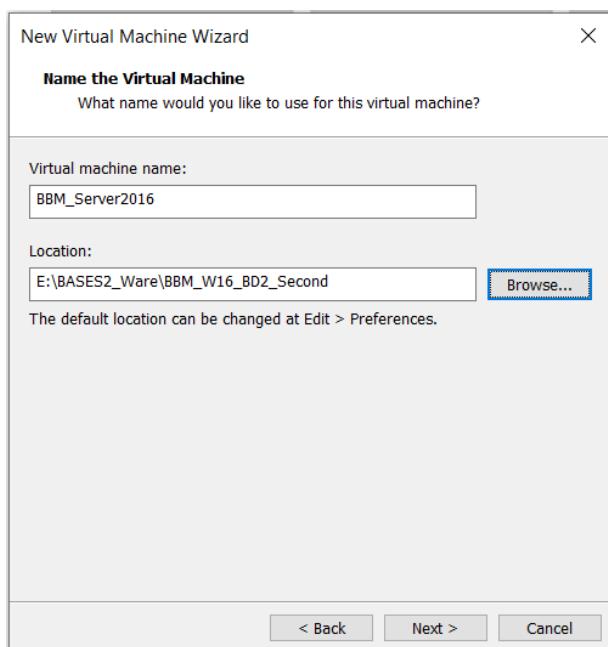


En la siguiente nos pide clave de producto, no dice la versión de Windows (Datacenter en este caso). En la parte de personalización de Windows pongo de nombre las iniciales de los apellidos y el nombre (**BBM** en mi caso), y como contraseña la más básica (*Abcd1234.*) para simular un entorno de trabajo:

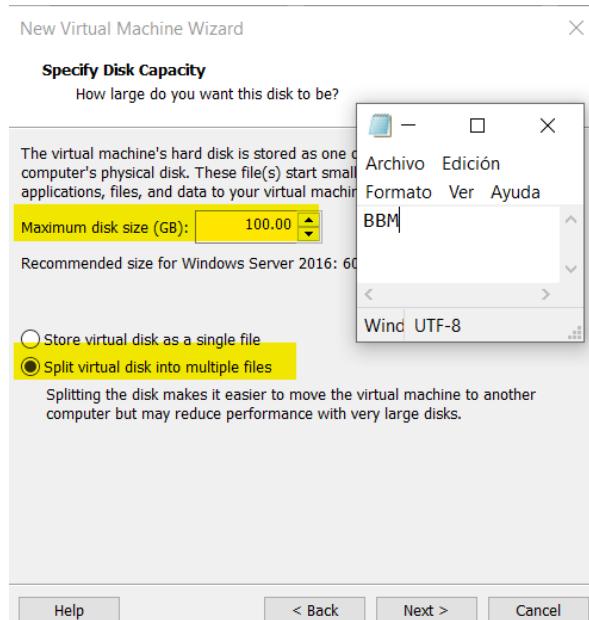


NOTA: Como no tenemos una clave activada, nos muestra un mensaje y le damos a **Yes**.

La siguiente ventana es para ponerle el nombre a nuestra máquina (en la cual intentaremos que todo tenga nuestro nombre) y una ubicación (en este caso el disco duro). Esta ubicación es una carpeta física:

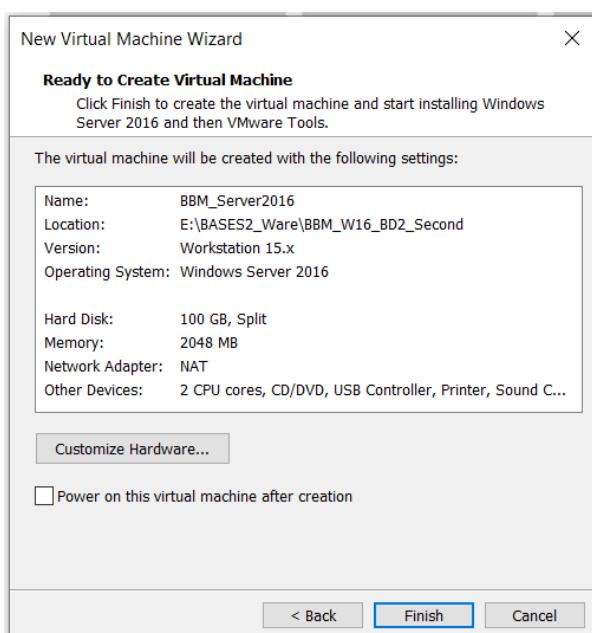


La siguiente ventana me muestra el tamaño del disco:

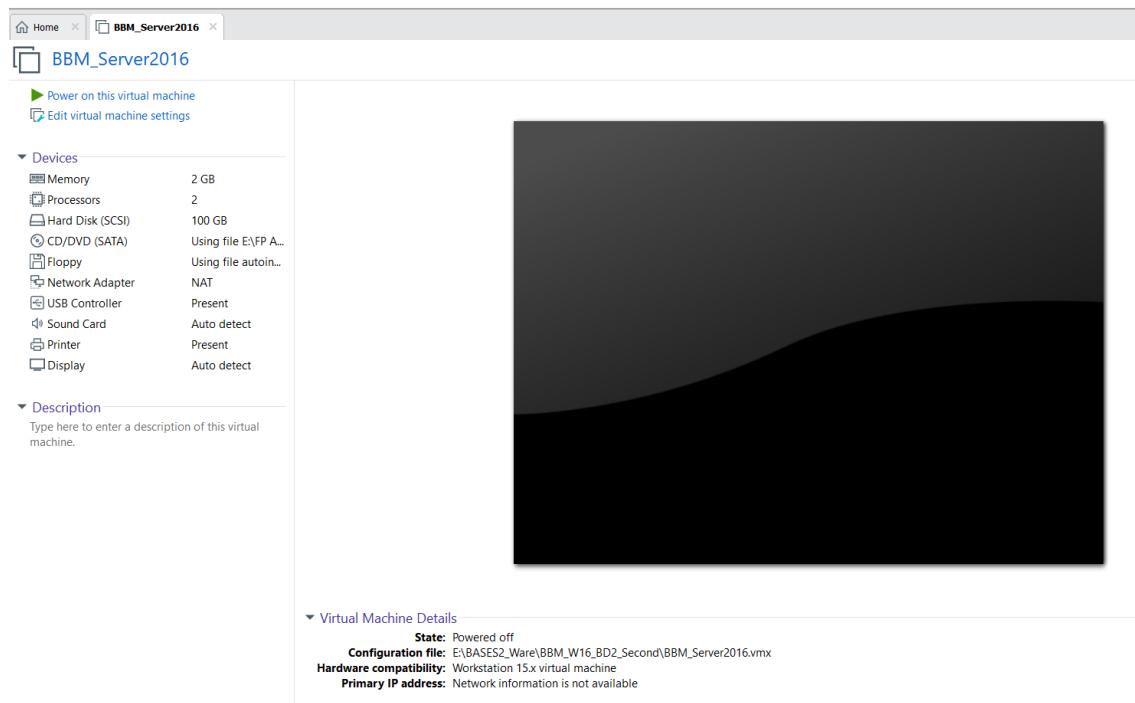


Como el proyecto se realiza a lo largo del curso, le daremos 100GB al menos porque luego aumentar el disco suele dar muchos problemas. En realidad yo reservo 100GB, pero al principio no ocupa 100, y va creciendo a medida que lo vamos llenando. En las opciones, cambio a la opción **Store virtual disk as a single file**. Le doy a *Next*:

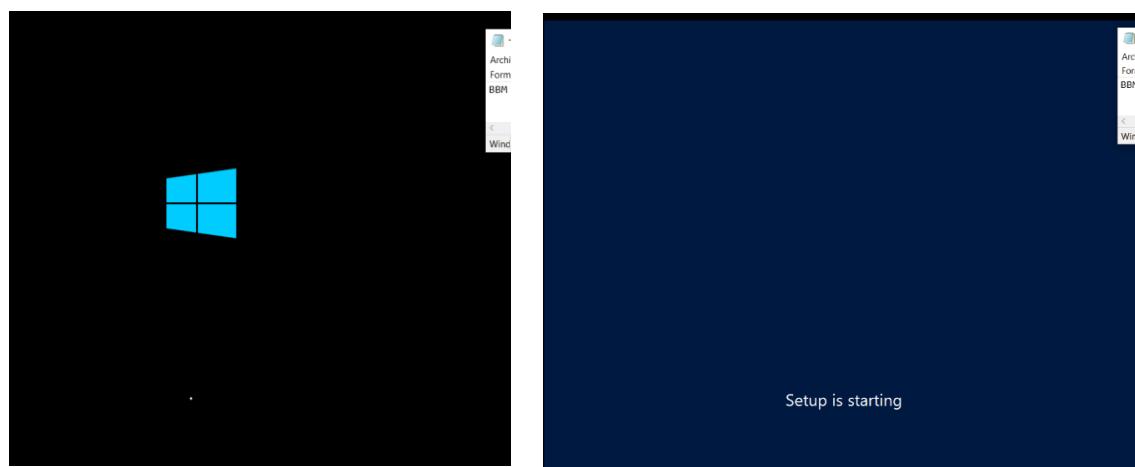
En la siguiente ventana, tendremos una especie de resumen y le damos a *Finish*.



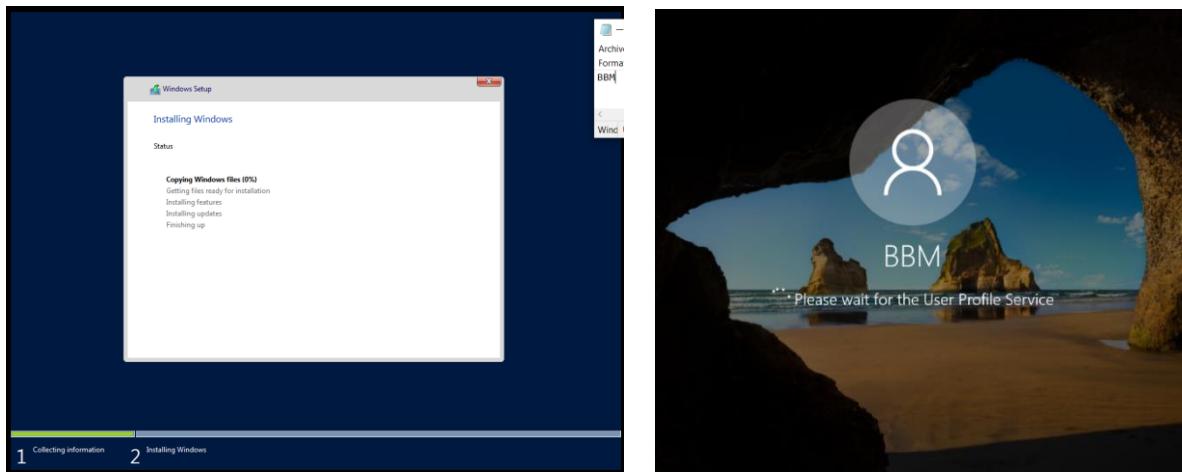
La presentación de la máquina creada se vería de esta manera:



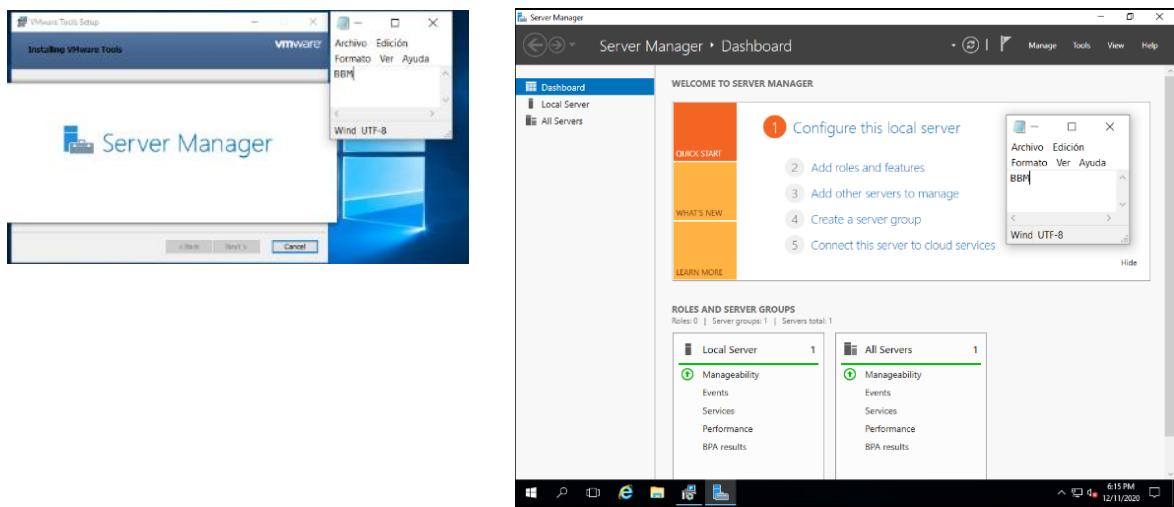
Para crear la máquina de Windows Server 2016, comienza la carga:



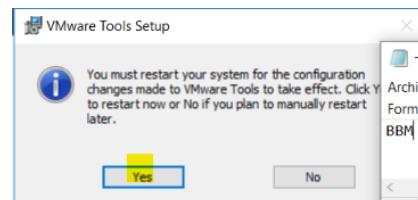
Empieza la instalación con recopilación de la información. Una vez acaba, inicia la carga del Sistema Operativo



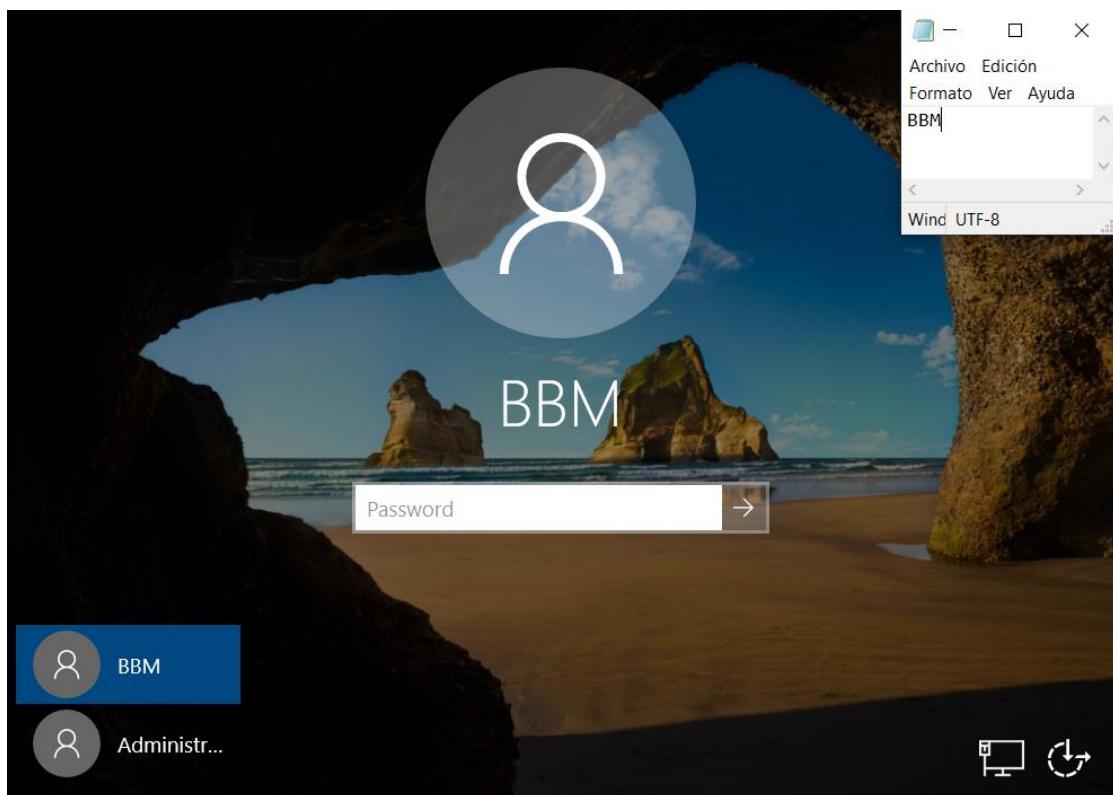
Cuando se inicia, se van a instalar las opciones de Server Manager, quedando de la siguiente manera:



NOTA: Cuando acaba, nos pide reiniciar. Le decimos que sí (Yes)

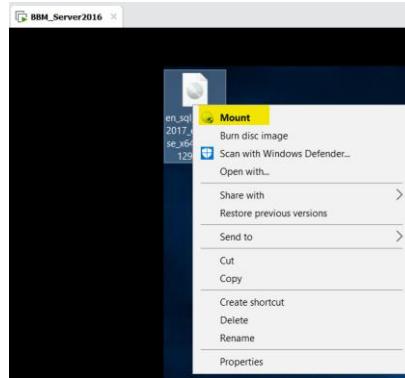


Cuando acaba de instalar y reiniciar, vemos la pantalla de la siguiente manera:

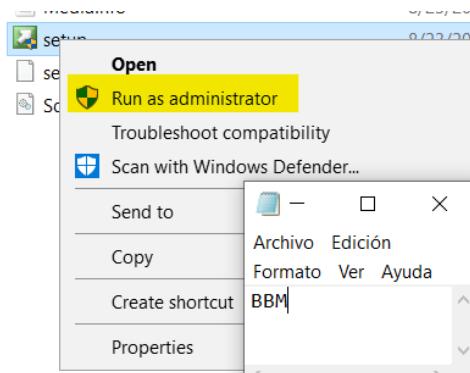


Instalación de SQL Server 2017

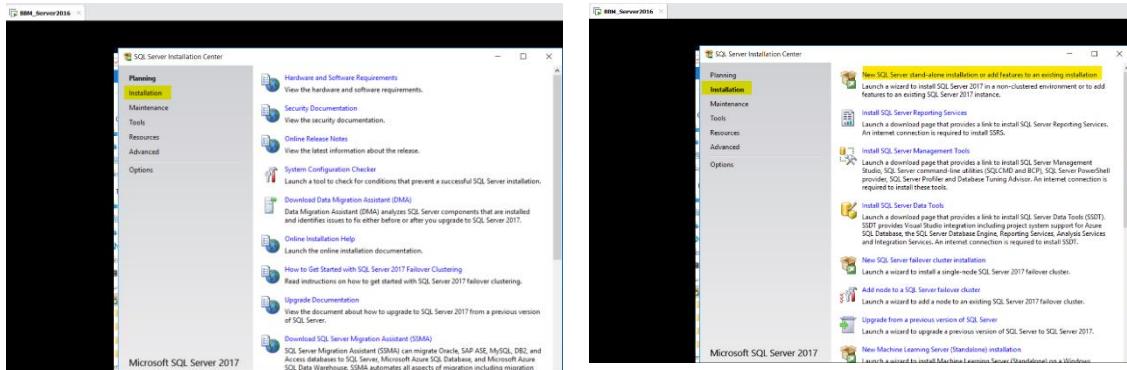
Lo primero que hacemos en la instalación es hacer botón derecho sobre la ISO y clicar en Mount. Al igual que con el Windows 10, la versión es *SQL Server 2017*.



Nada más montar, iniciamos como administrador, clicando con el botón derecho sobre el ícono *setup* y dando a **Run as administrator**. Aceptamos la instalación.



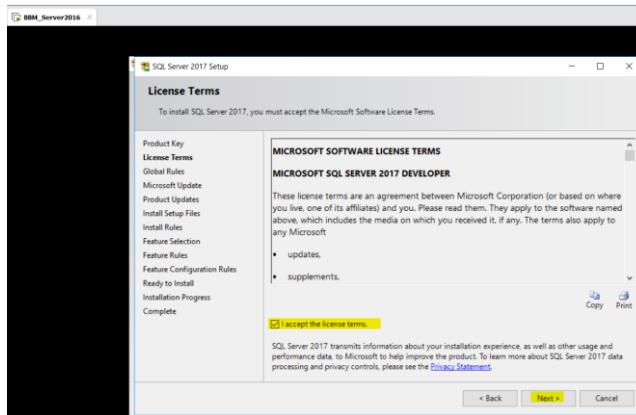
La siguiente pantalla que sale es un panel de ayuda. Son links que en ocasiones valen para conectar en SQL Server. En este caso, lo que se va a usar es *Installation*. Dentro de todas las opciones que nos encontramos, esta vez vamos a utilizar *New SQL Server stand-alone or add feature to an existing installation*. Podemos distinguir entre una instalación stand-alone (único equipo) o un clúster (agrupación). En este caso instalaremos un equipo (stand-alone) y le añadiremos algunas características.



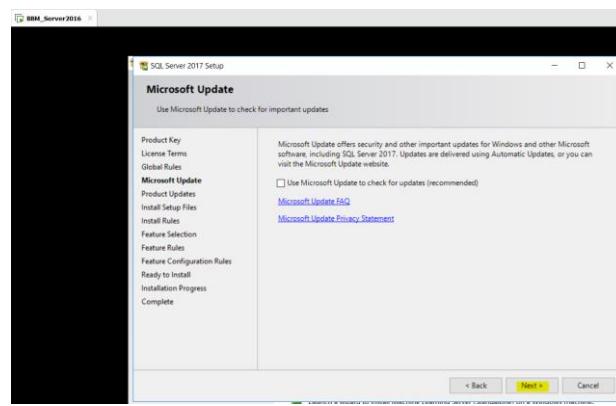
Se abre una nueva ventana donde me dice que especifique un tipo de edición. *Evaluation* sería la opción trial; *Express* es la opción gratuita que solemos utilizar en pequeñas empresas, y *Developer* es la que tiene todas las características sin ponerlas en producción. En caso de haber pagado la licencia, elegiríamos la opción de poner una key. En este caso elegimos la opción *Developer*.



Aceptamos los términos de la licencia y le damos a continuar.

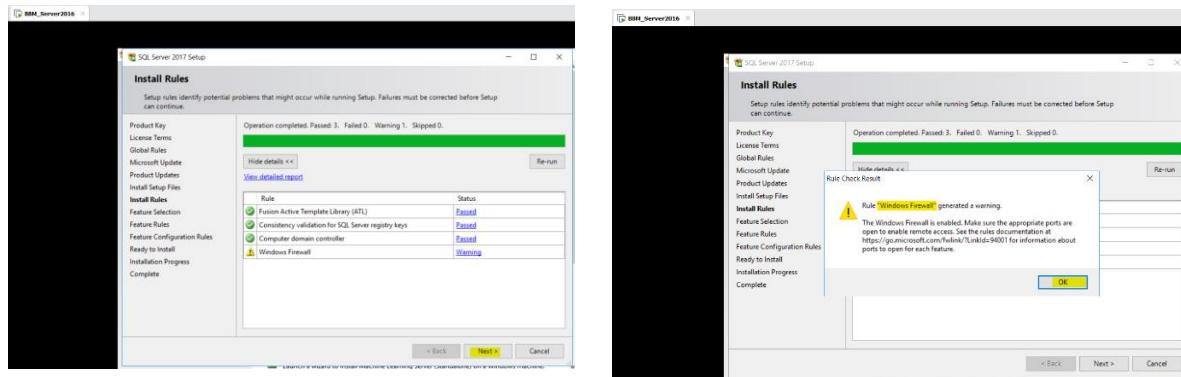


La siguiente ventana hace como una previa de lo que tenemos instalado en nuestro equipo, y nos pregunta si queremos realmente que busque actualizaciones automáticamente. En este caso no nos interesa tal cosa, así que le damos a continuar sin clicar en el botón de check.



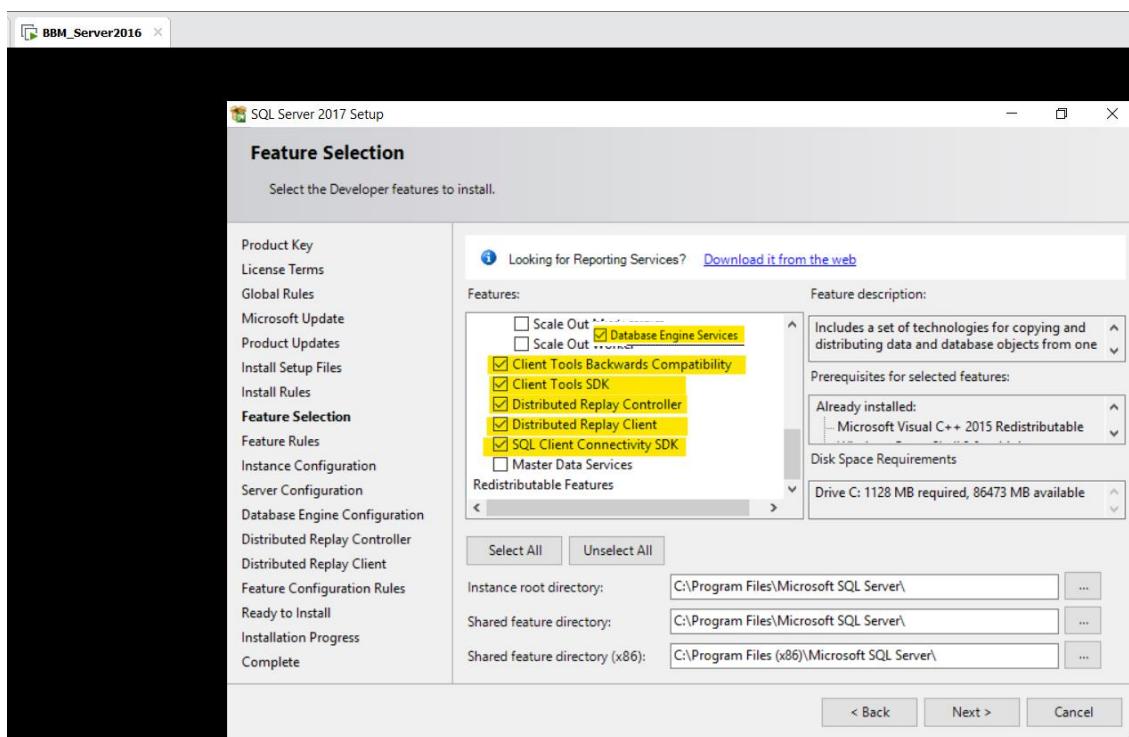
La siguiente es una ventana donde empezará con la instalación. Los check verdes quieren decir que la instalación se ha completado de manera satisfactoria, rojo un

error, y amarillo, un *warning* que nos avisa de que hay algo que no va a funcionar correctamente hasta que se solvete (en este caso el firewall). En la parte de *Status*, tenemos un link, que da una información adicional en una ventana nueva sobre el evento que seleccionemos. En este caso voy a utilizar el Warning para que quede reflejado. Tras leer la información que me interese, le daría a *Next*:

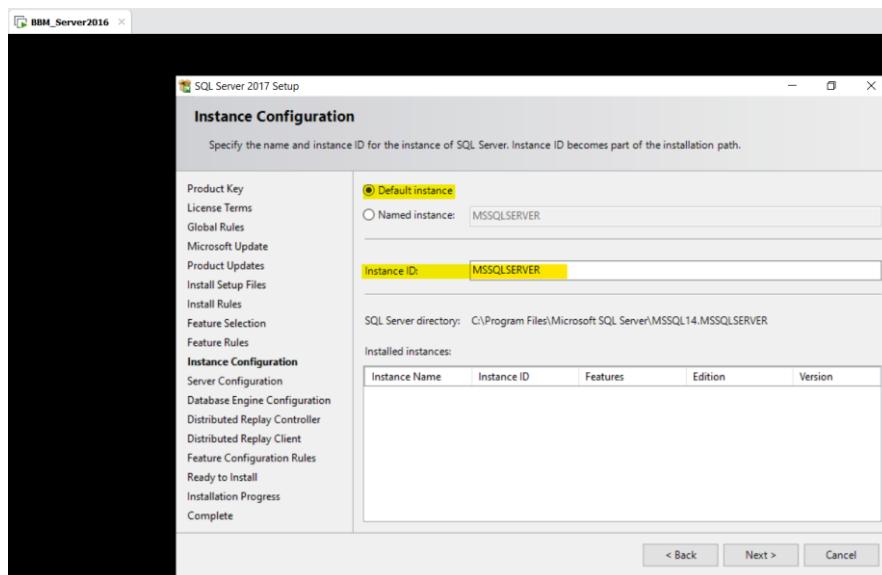


NOTA: En los siguientes apartados este *WARNING* será *importante*, pues tendremos que jugar en sus opciones para interconectar las máquinas

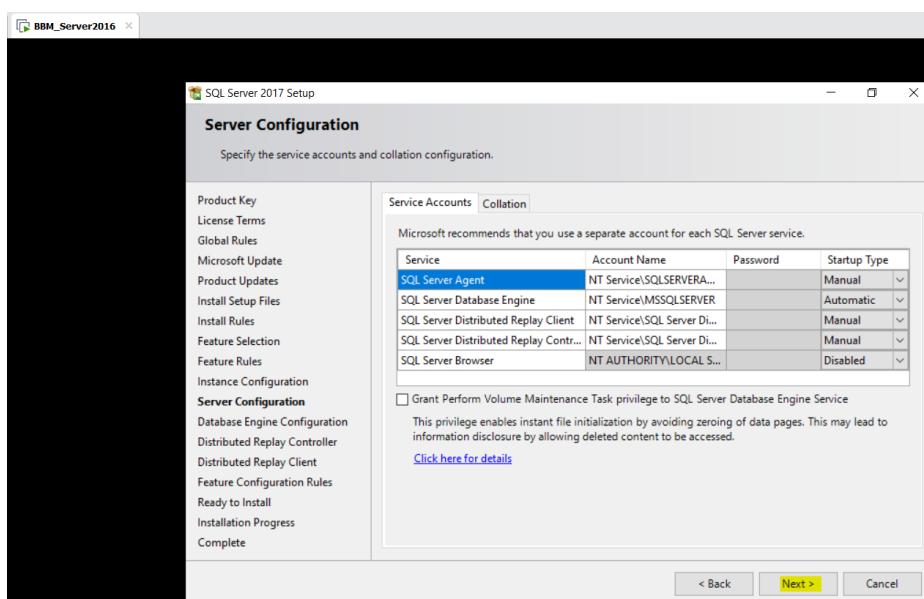
En la siguiente ventana, elegiremos qué elementos quiero elegir. Elegiremos los servicios de bases de datos (*Database Engine Services*). También elegiré en las características compartidas todas las que tengas que ver con cliente: *Client Tools Connectivity*, *Client Tools Backwards Compatibility*, *Client Tools SDK*, *Distributed Replay Controller*, *Distributed Replay Client*, *SQL Client Connectivity SDK*. Siempre se pueden cambiar estas características a posteriori. Todos los directorios los dejo por defecto.



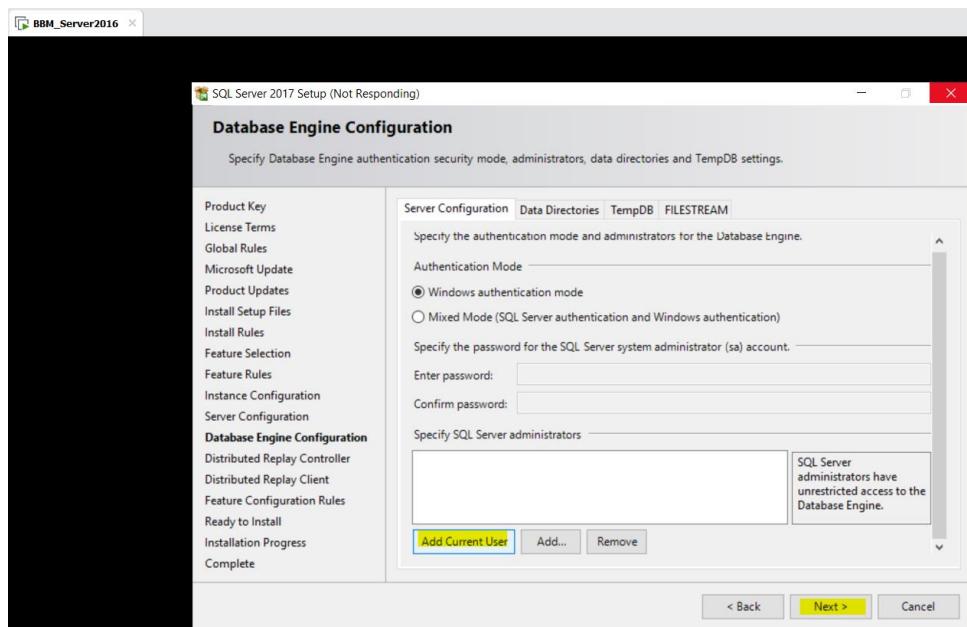
En la siguiente parte, configuraremos una instancia. Lo que aquí llama una instancia es un servidor de base de datos. La primera es la instancia por defecto. Si fuese la segunda, elegiría *Named instance*. Elijo *Default instance* y le doy a Next:



Al darle a siguiente nos van a aparecer los servicios. Los que aparecen son los que necesitamos. El más importante es el SQL Server Database Engine (el sistema de gestión). Todo se instalaría en cuentas de dominio, pero para nuestras pruebas queda por defecto:

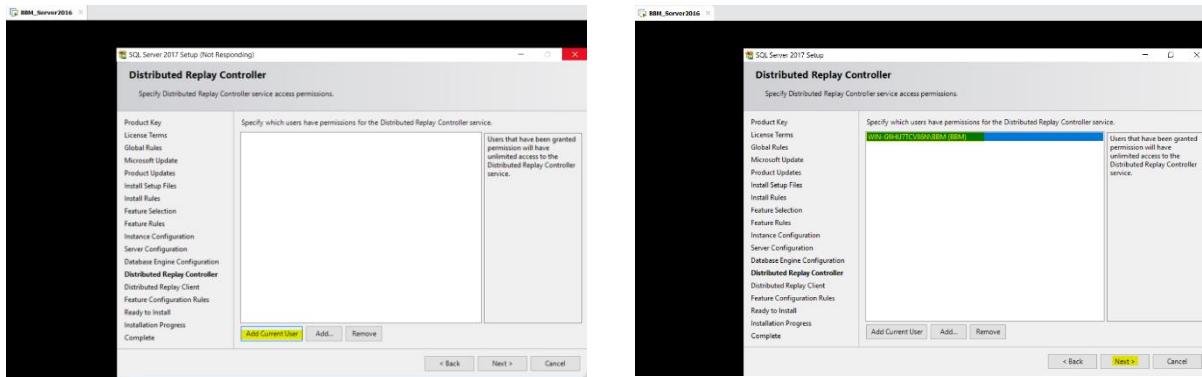


Lo siguiente que vamos a configurar es el modo de autenticación. En este caso dejamos la opción de Windows, y le vamos a dar a añadir usuario actual (*Add Current User*).

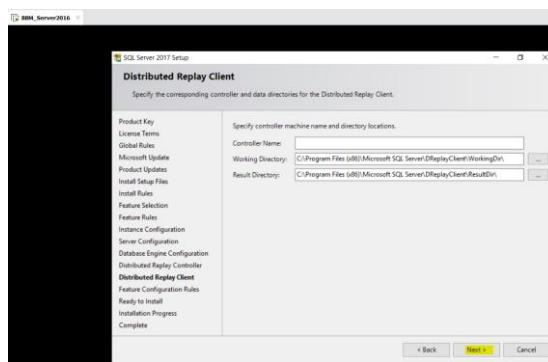


Vemos que al cabo de unos segundos aparece nuestra cuenta. Si exploramos el resto de las pestañas, vemos que podríamos variar los directorios, modificar la base TempDB, o subir archivos

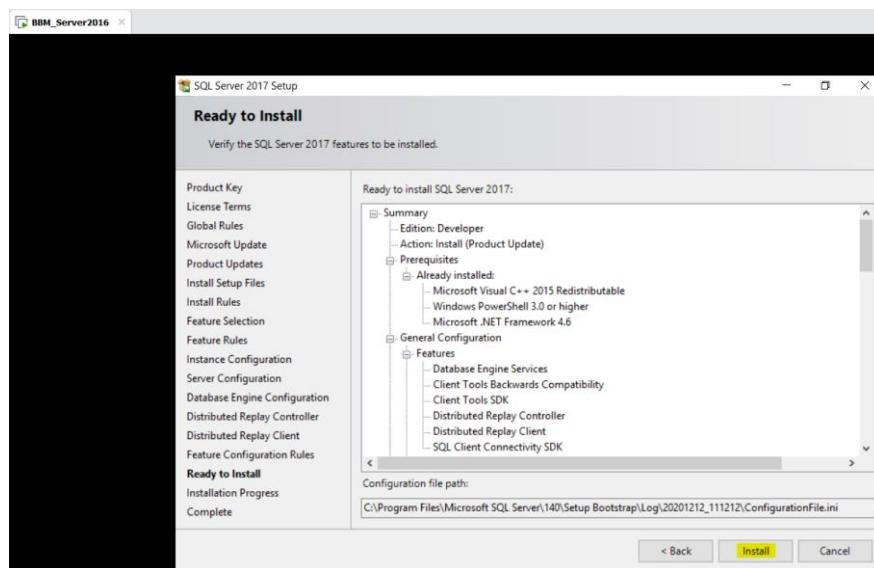
En la siguiente venta le especificamos el usuario con permiso en *Add Current User*:



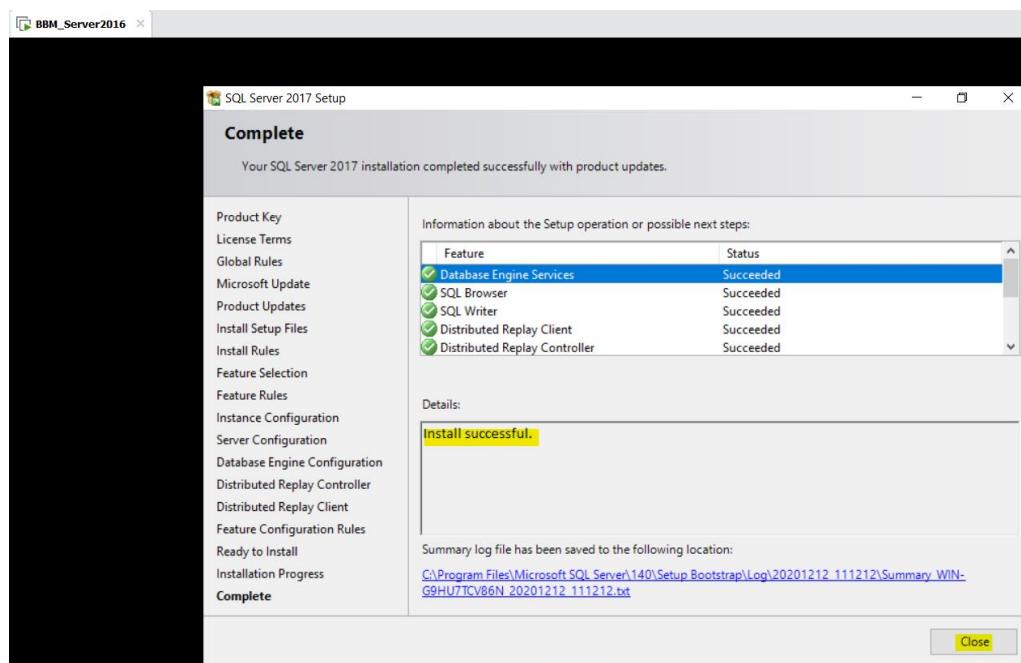
La siguiente ventana le damos a Next



En esta ventana nos da un resumen antes de instalar. Comprobamos que todo está ok e instalamos SQL SERVER. Como apunte, el *Configuration Path* sirve para las instalaciones desatendidas, cosa que explicaré más adelante.



La última página es un resumen de todo lo instalado. Le damos a *close* y como en todos los programas reiniciamos el ordenador.



Configurar Firewall de Windows para permitir el acceso a SQL Server

Ahora comenzamos a meternos en interconectar los equipos cliente y servidor. Para ello hay que salvar el primer escollo que se presenta: el **Firewall**. Para que se pueda conectar perfectamente, vamos a permitir tres cosas:

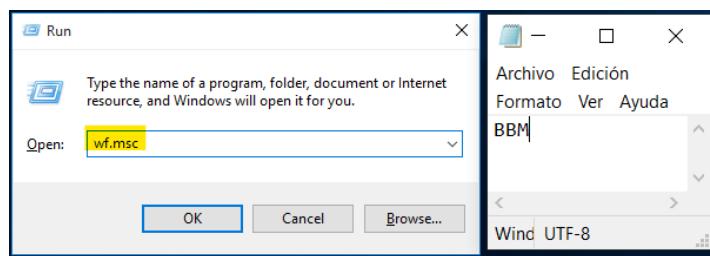
- El puerto **1433** en protocolo TCP/UDP
- El puerto **1434** en protocolo TCP/UDP, que puede ser necesario para el servicio **Explorador de SQL Server** cuando se utilizan instancias con nombre
- La red en la que van a estar encuadrados los tres equipos. En mi caso en la **192.168.0.0/29**, que se verá más adelante

Configuración del Firewall en Windows 10

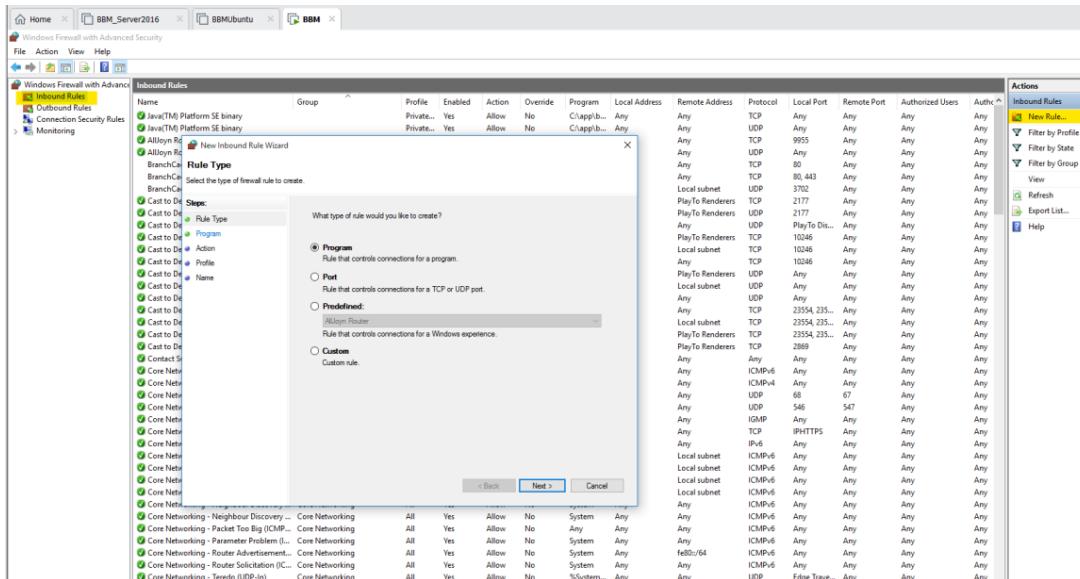
Por da un orden, primero creo las reglas del puerto. Esto es realmente todo igual, aunque por dejarlo todo bien documentado y explicado, voy a repetir los pasos en todo.

Creación de regla Inbound para permitir el puerto 1433

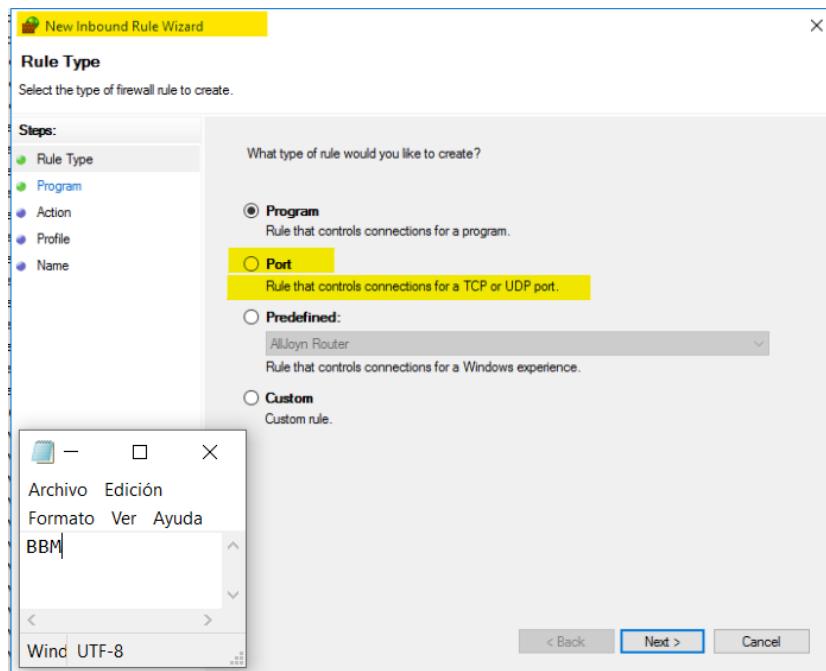
Pulsamos la tecla **R** + **R** y escribimos **wf.msc** (*Windows Firewall*) para acceder a las reglas del Firewall:



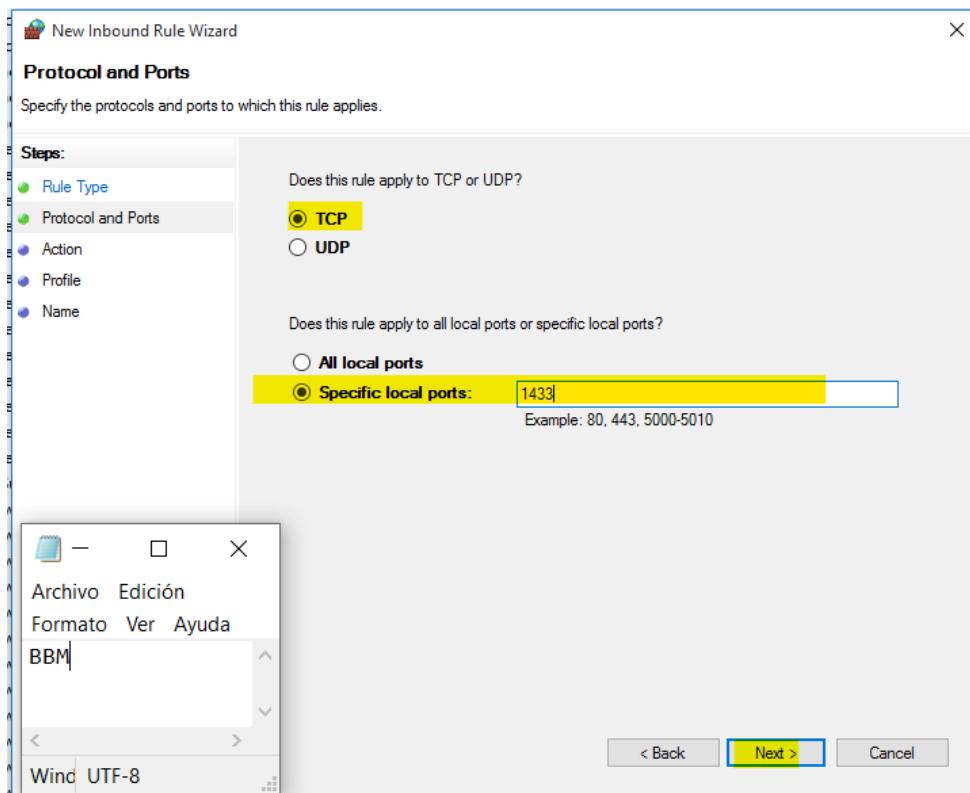
Una vez dentro, seleccionamos **Inbound Rules** (Reglas de entrada)



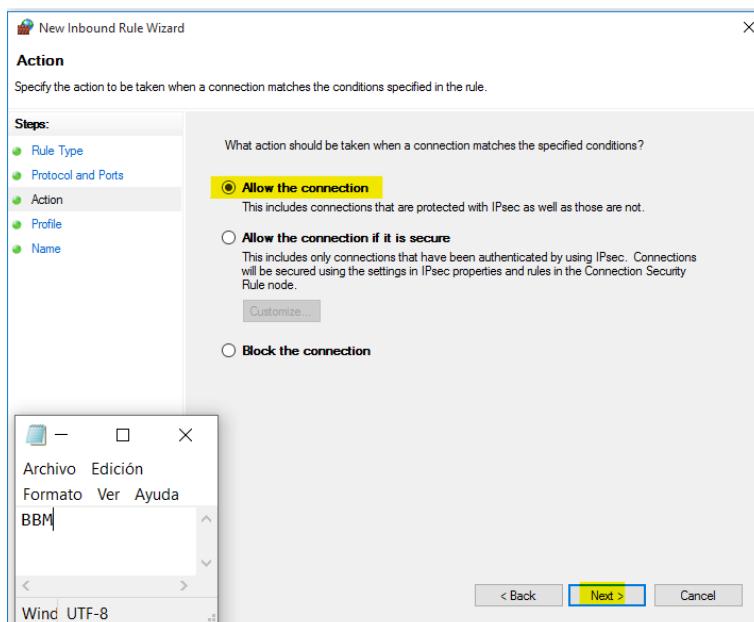
Cuando se abre nos da a elegir entre cuatro opciones: Program, Portm Predefined, Custom. En este caso vamos a elegir Port para especificar el puerto que queremos que el SQL lo permita:



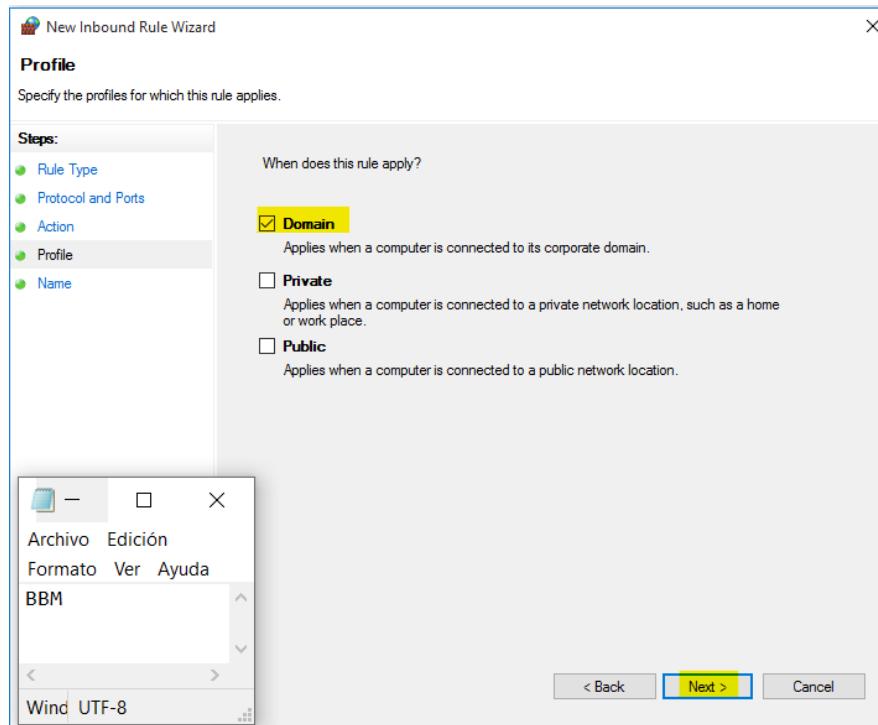
En la siguiente le especificamos el protocolo, tanto TCP como UDP vamos a hacer. Se hace uno por uno, yo decidí empezar por TCP:



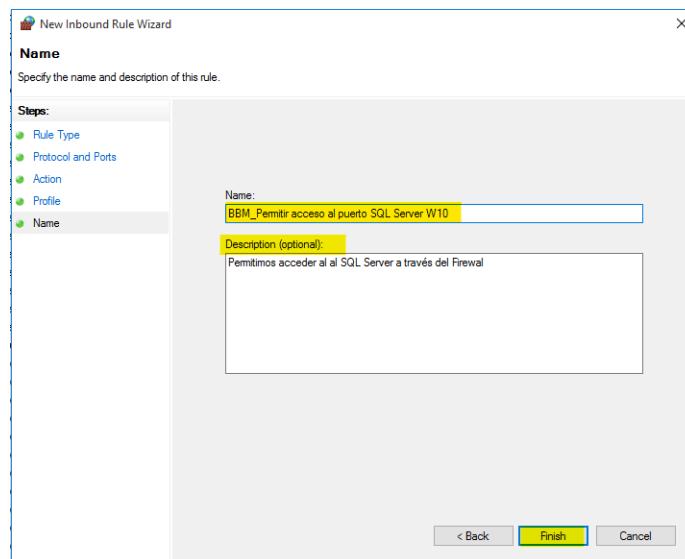
En la siguiente pestaña podemos elegir entre permitir la conexión, permitirla sólo si es segura o bloquearla. En este caso, permitimos la conexión (*Allow the connection*):



En la siguiente ventana elegimos entre dominio, una red privada o pública. Elegimos **Domain** porque vamos a conectar entre los tres equipos metiéndolos en un dominio.



Por último, ponemos un nombre a la regla y una descripción (opcional) y finalizamos pulsando en Finish

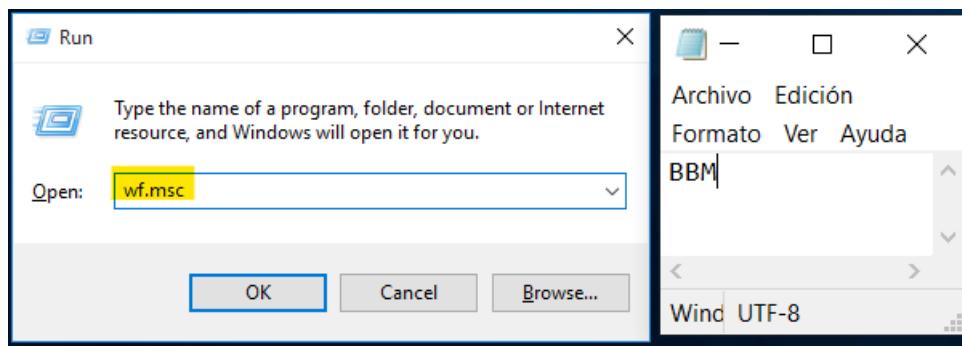


Revisamos que la regla esté creada:

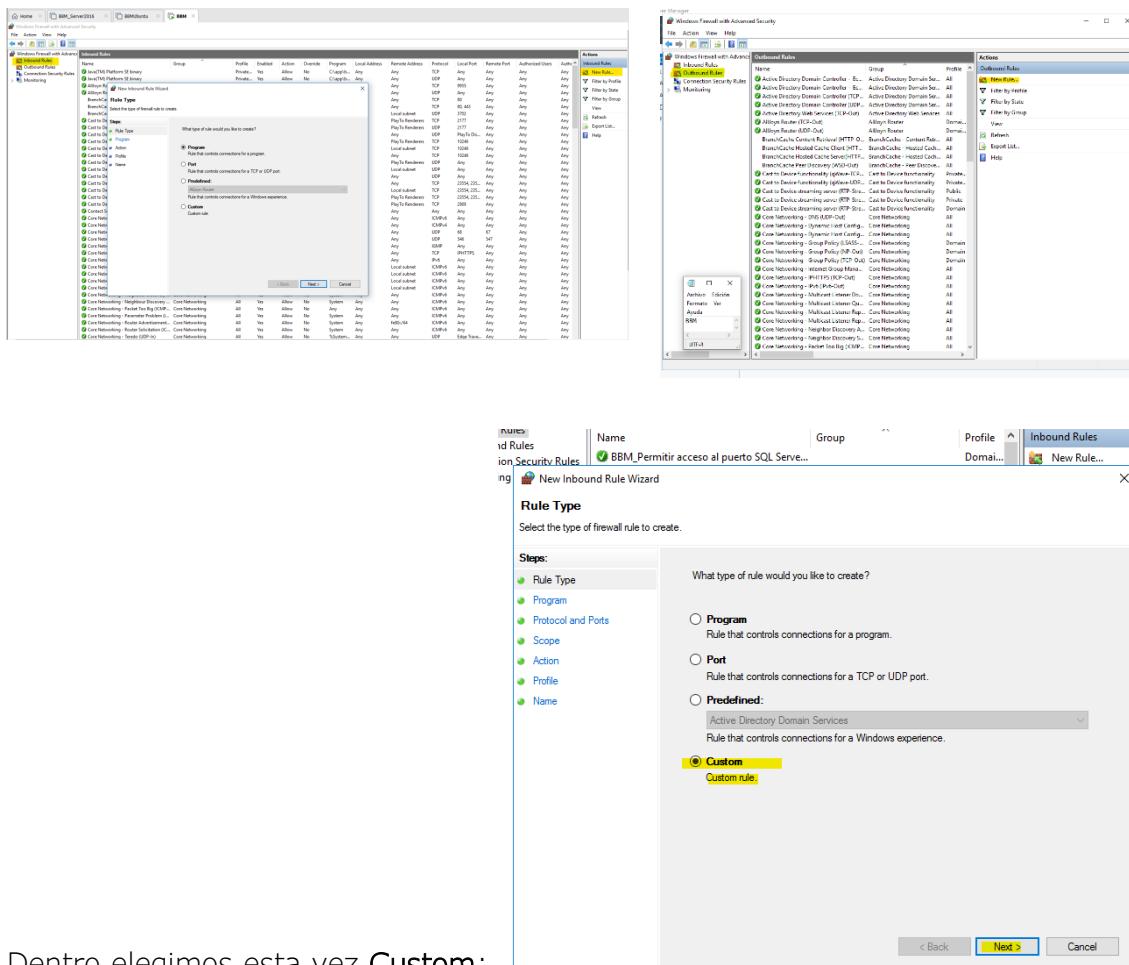
Name	Group	Profile	Enabled	Action	Override	Program	Local Address	Remote Address	Protocol	Local Port	Remote Port	Authorized Users	Authentication
BBM_Permitir acceso al puerto SQL Server W10	Default	Domain	Yes	Allow	No	Any	Any	Any	TCP	1433	Any	Any	Any
Java(TM) Platform SE binary	Private	Private...	Yes	Allow	No	C:\app\bbm\bin\java.exe	Any	Any	TCP	Any	Any	Any	Any
Windows Firewall with Advanced Security	Public	Public	Yes	Allow	All	Windows Firewall	Any	Any	Any	Any	Any	Any	Any

Creación de regla Inbound/Outbound para permitir el puerto 1434

En este caso repetiríamos los mismos pasos que la anterior, pero a medida que fui haciendo pruebas, me di cuenta de que podía agregar los dos puertos a la vez, así que repetí todo lo anterior. Primero  + R para acceder al Firewall:

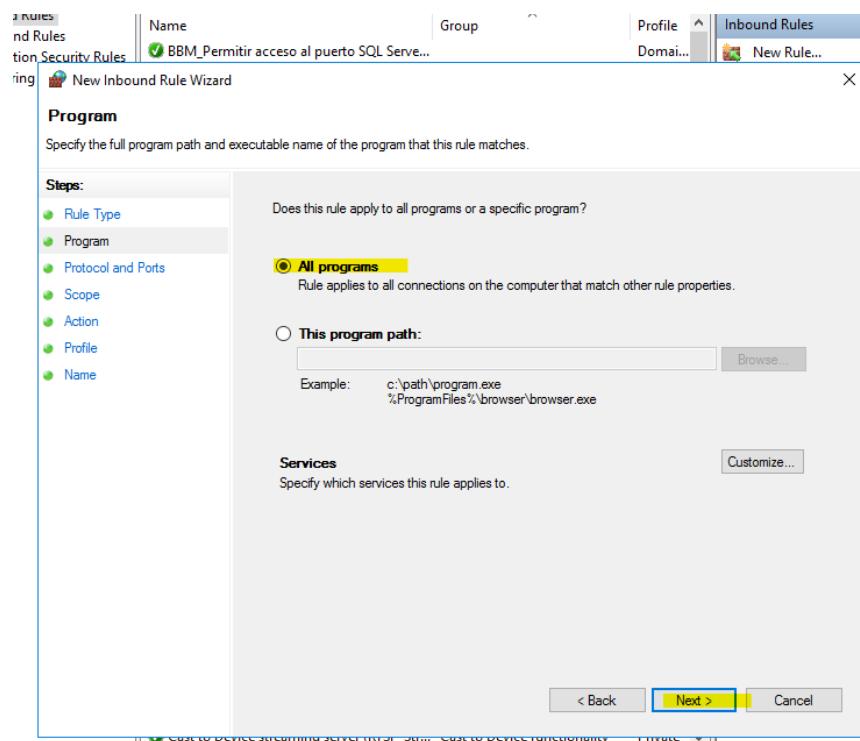


En la pantalla con las Inbound/Outbound Rules creadas, le damos a New Rule...

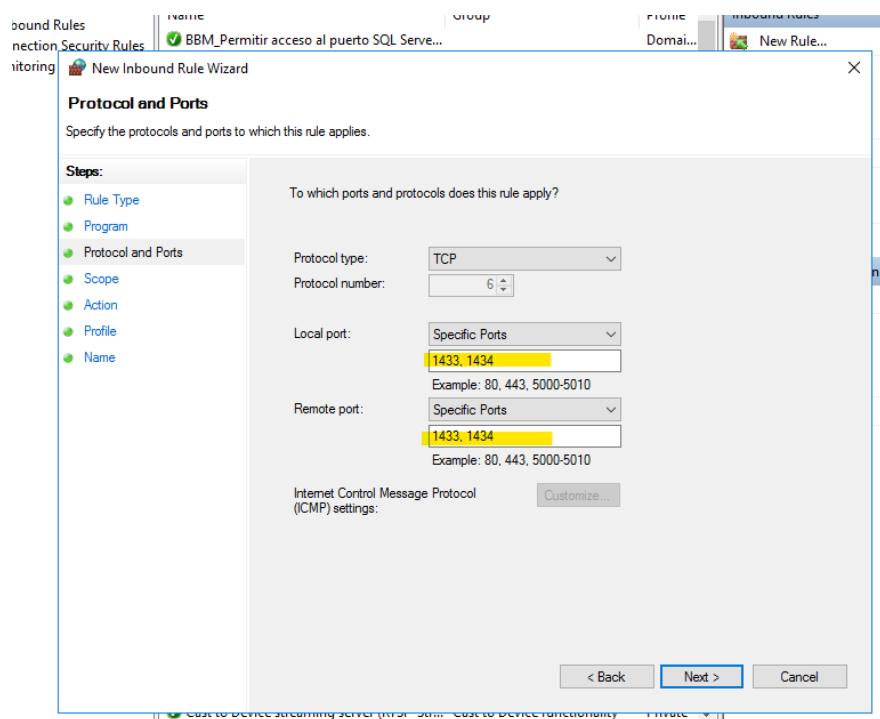


Dentro elegimos esta vez Custom:

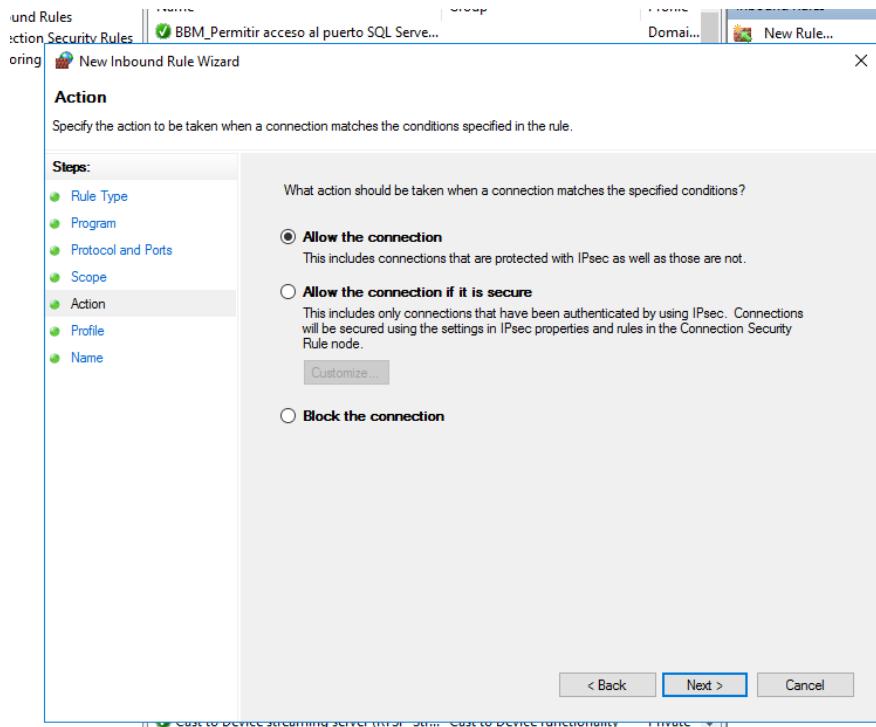
En la siguiente ventana, podría haber elegido una ruta específica, pero tuve miedo de que fallase toda la regla, así que puse la opción All programs.



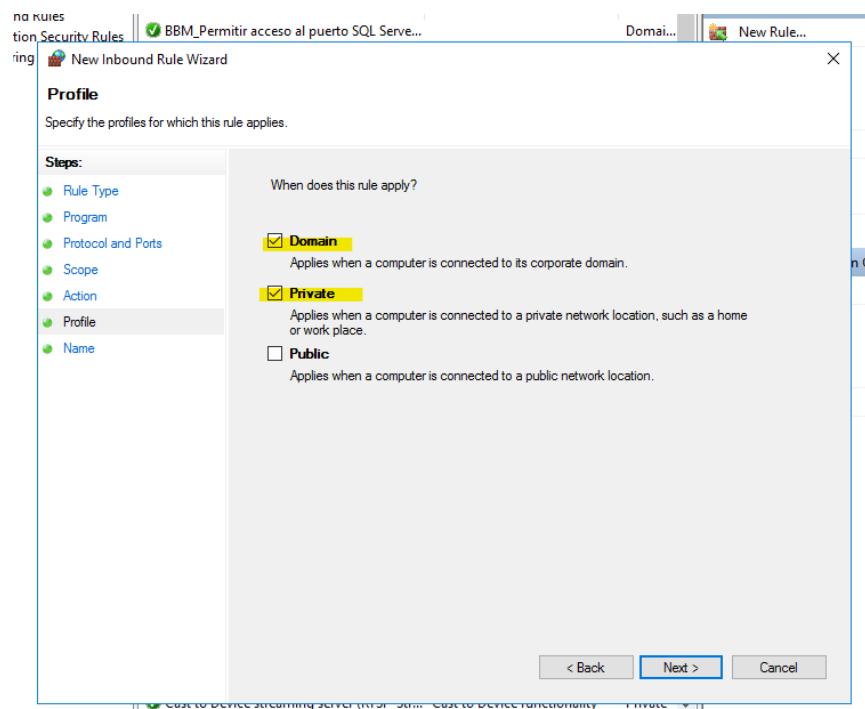
En la siguiente ventana es donde me di cuenta que podría elegir más de un puerto y permitir en ambos sentidos los puertos, tanto local como remotamente, así que los introduce separados por una coma, protocolo **TCP**:



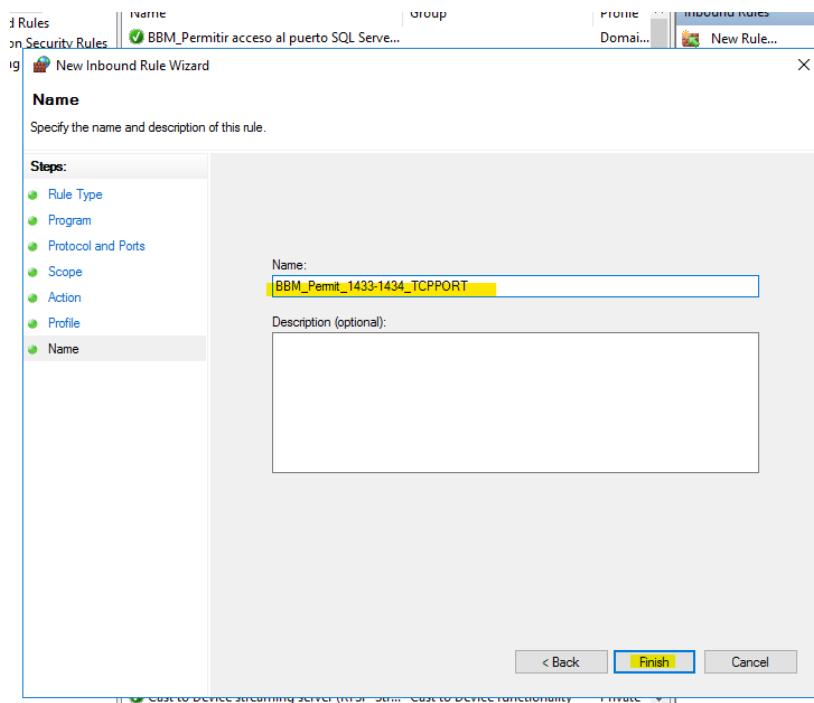
Al igual que en la anterior, Allow the connection:



En la siguiente ventana, antes solo hablaba de dominio, pero al ser una red privada por la que pasa la información, elegimos **Domain** y **Private**:



En la última le ponemos el Nombre y le damos a Finish.



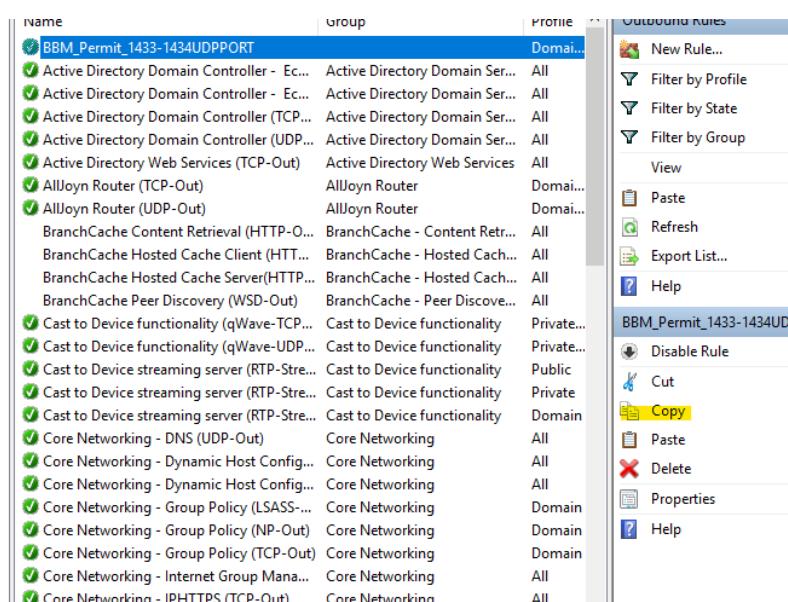
El resultado

The screenshot shows the Windows Firewall with Advanced Security - Inbound Rules list. The rule 'BBM_Permit_1433-1434TCPORT' is listed under the Inbound Rules category.

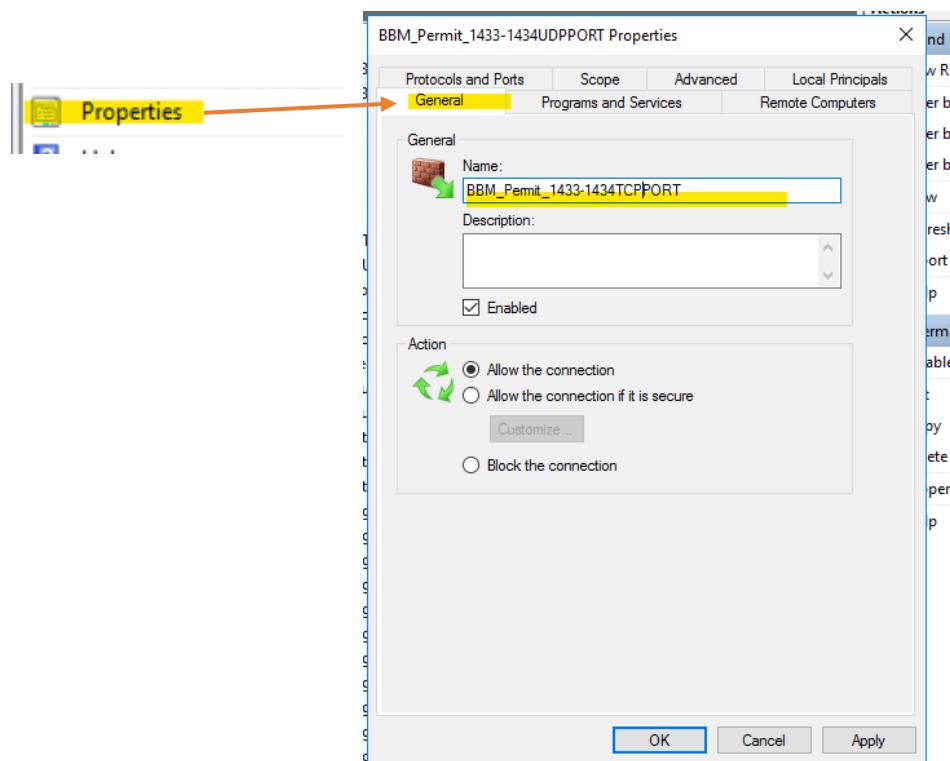
NOTA: La regla de 1433 que expliqué arriba la he borrado porque no quería duplicidades

Para la regla UDP lo que hice fue copiar la regla que había creado con **copy-paste**:

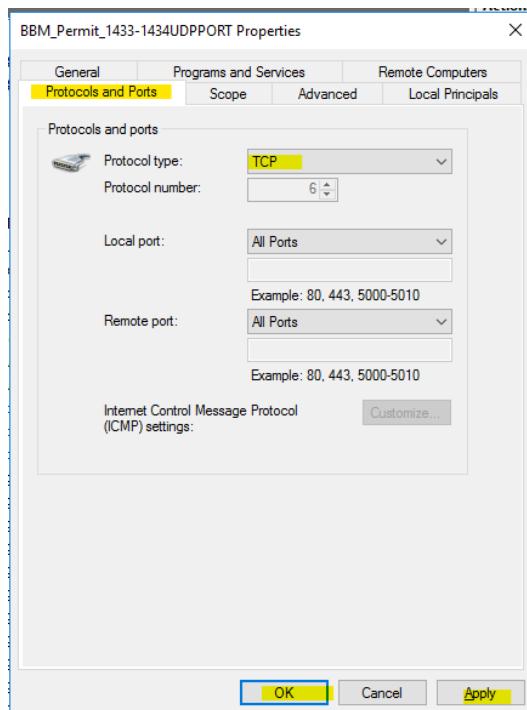
The screenshot shows the Windows Firewall with Advanced Security - Inbound Rules list. A context menu is open for the rule 'BBM_Permit_1433-1434UDPORT'. The 'Paste' option is highlighted in yellow.



Tocando en Properties se abre una ventana donde en la pestaña General, podemos cambiarle el nombre.

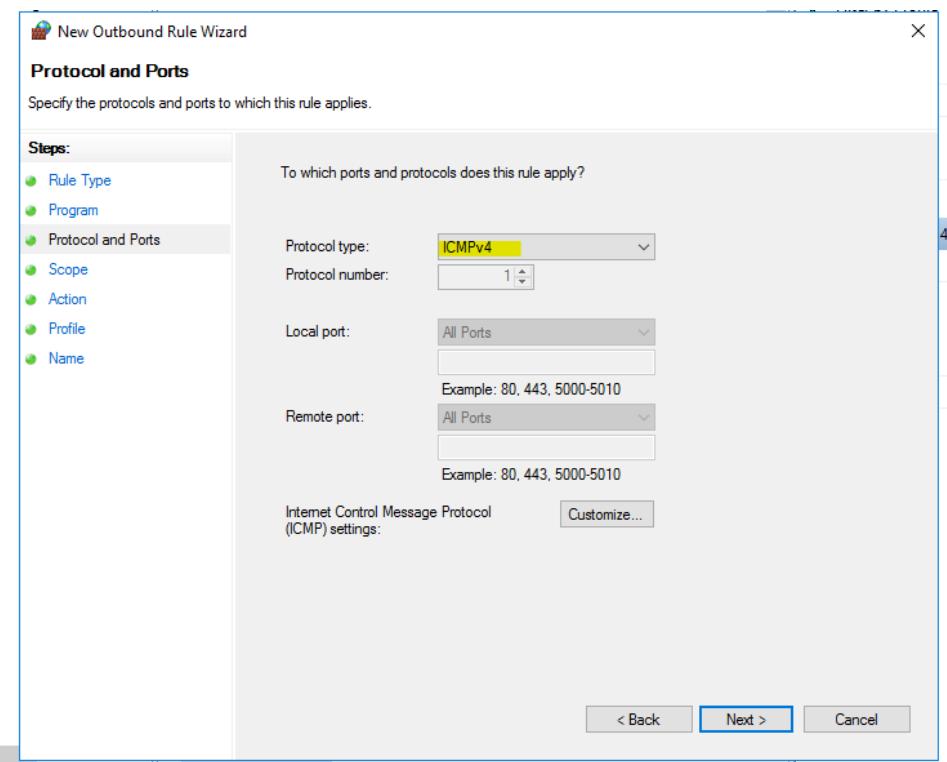


En la pestaña Protocols and Ports se cambia el puerto a TCP. Apply y OK y el cambio ya quedaría hecho

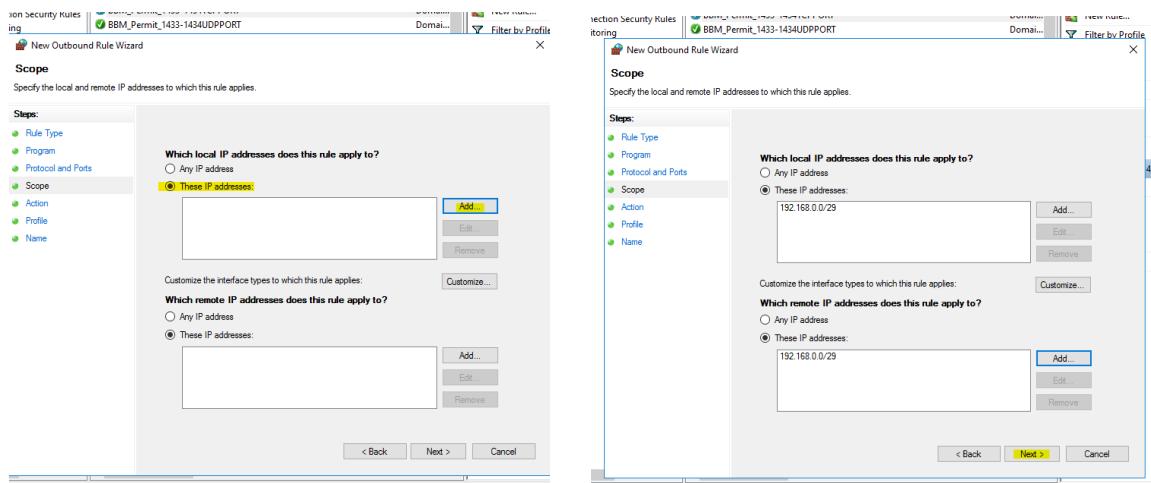


Creación de regla Inbound/Outbound para permitir la red

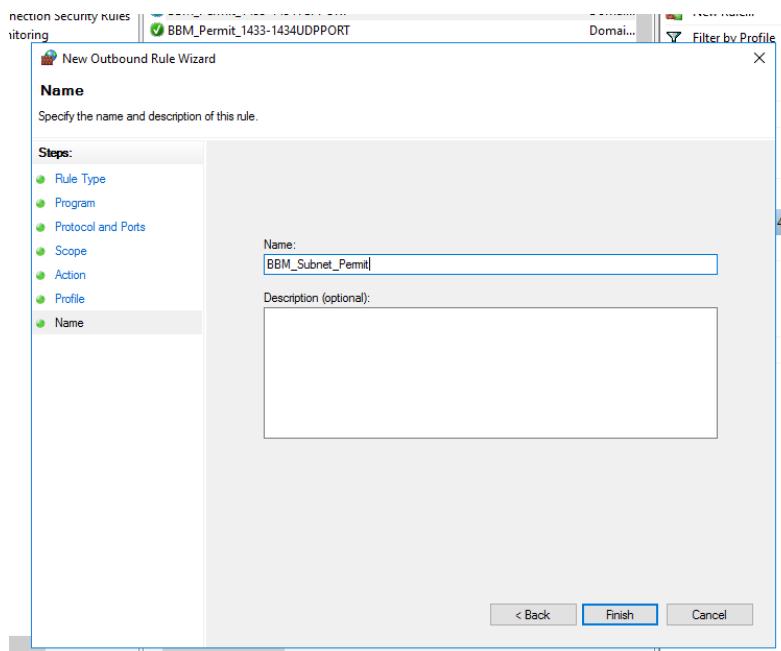
El método de creación básico ya quedó explicado, ahora dejo puestas las capturas donde cambia. En **Protocols and Ports**, elegimos de protocolo el **ICMPv4**, protocolo que trabaja en la capa 3 del modelo OSI y está relacionado con las IP.



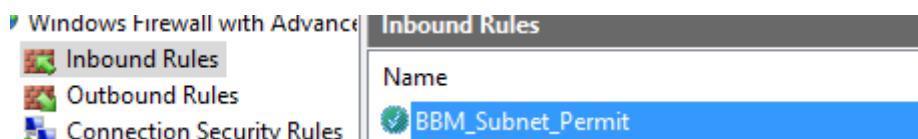
En la siguiente pestaña le decimos las IP que va a llevar. En mi caso, como hice una **subred**, agregué la subred 192.168.0.0/29.



Para finalizar, nombro la regla y acepto.



Dejo constancia de la regla finalizada:

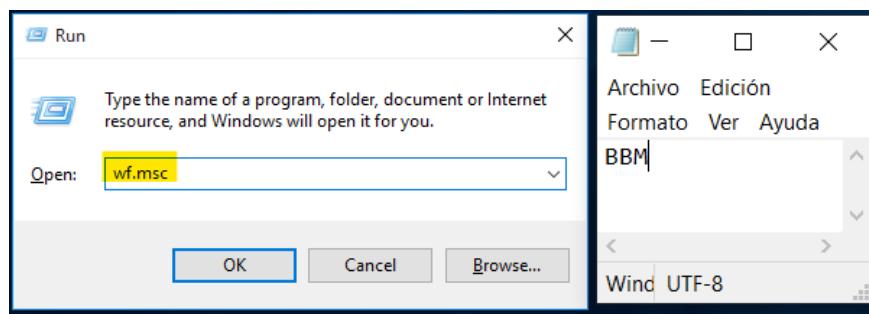


Configuración del Firewall en Windows Server 16

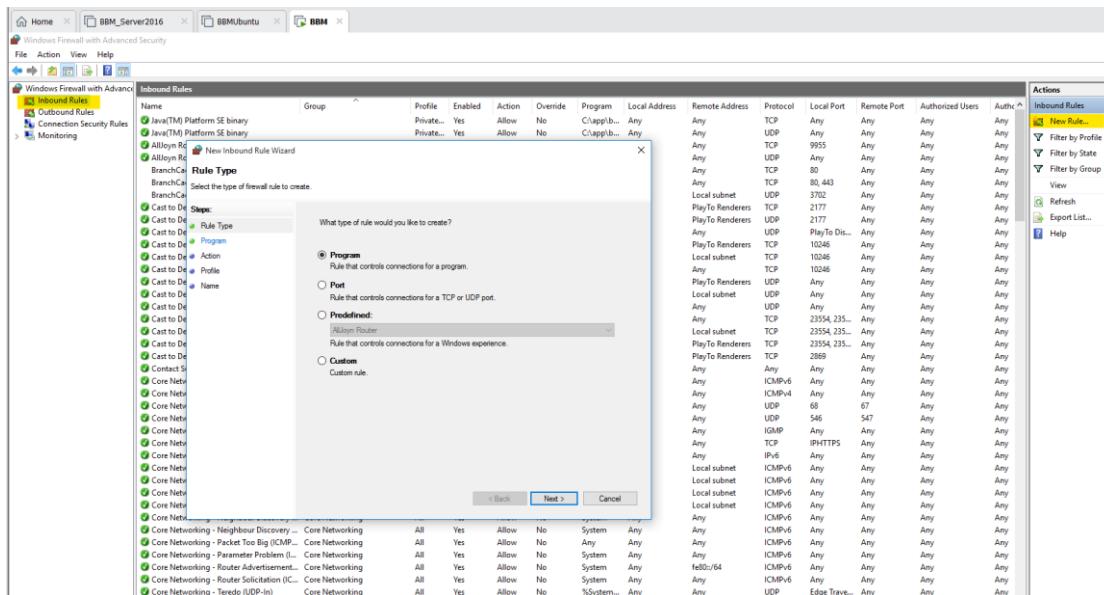
Por da un orden, primero creo las reglas del puerto. Esto es realmente todo igual, aunque por dejarlo todo bien documentado y explicado, voy a repetir los pasos en todo.

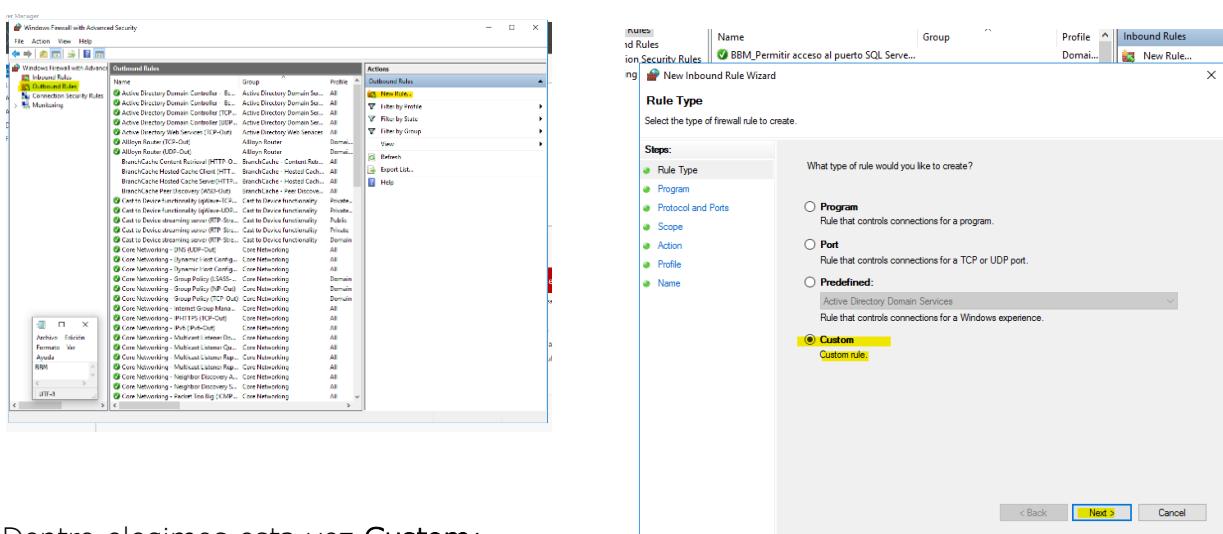
Creación de regla Inbound/Outbound para permitir el puerto 1434

En este caso repetiríamos los mismos pasos que la anterior, pero a medida que fui haciendo pruebas, me di cuenta de que podía agregar los dos puertos a la vez, así que repetí todo lo anterior. Primero **R** + **R** para acceder al Firewall:



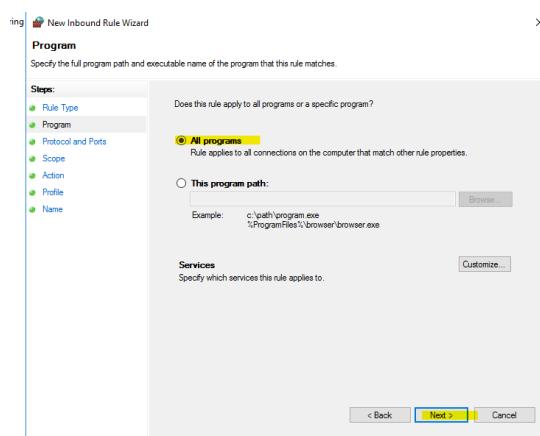
En la pantalla con las Inbound/Outbound Rules creadas, le damos a **New Rule...**



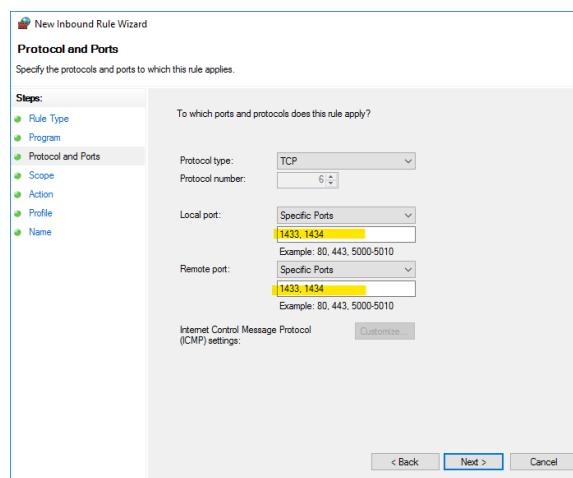


Dentro elegimos esta vez Custom:

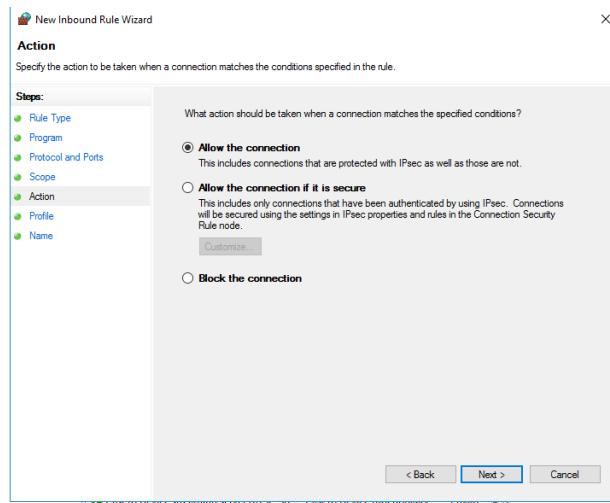
En la siguiente ventana, podría haber elegido una ruta específica, pero tuve miedo de que fallase toda la regla, así que puse la opción All programs.



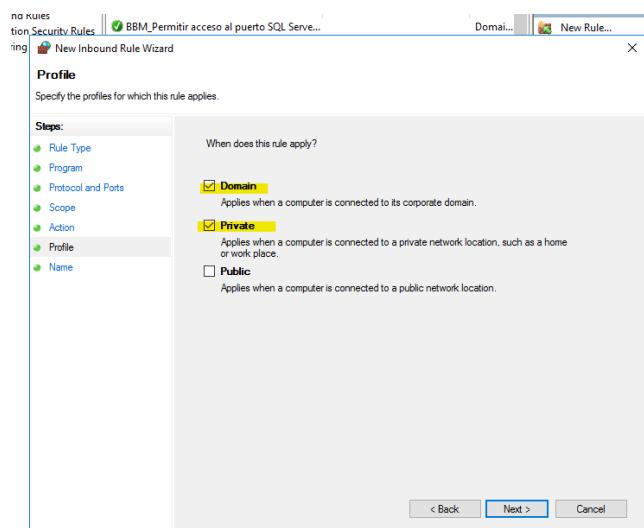
En la siguiente ventana es donde me di cuenta que podría elegir más de un puerto y permitir en ambos sentidos los puertos, tanto local como remotamente, así que los introduje separados por una coma, protocolo TCP:



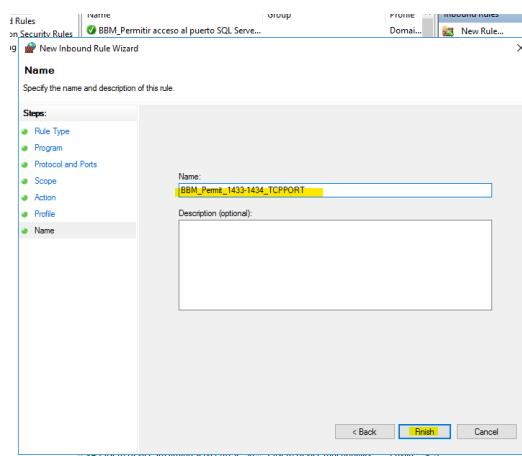
Al igual que en la anterior, Allow the connection:



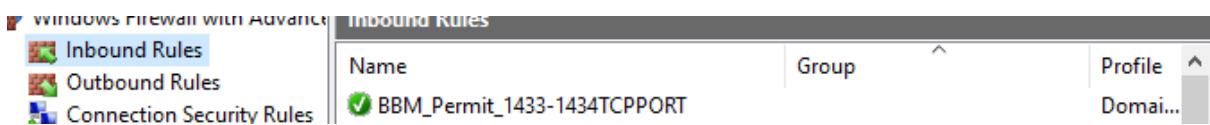
En la siguiente ventana, antes solo hablaba de dominio, pero al ser una red privada por la que pasa la información, elegimos **Domain** y **Private**:



En la última le ponemos el Nombre y le damos a **Finish**.

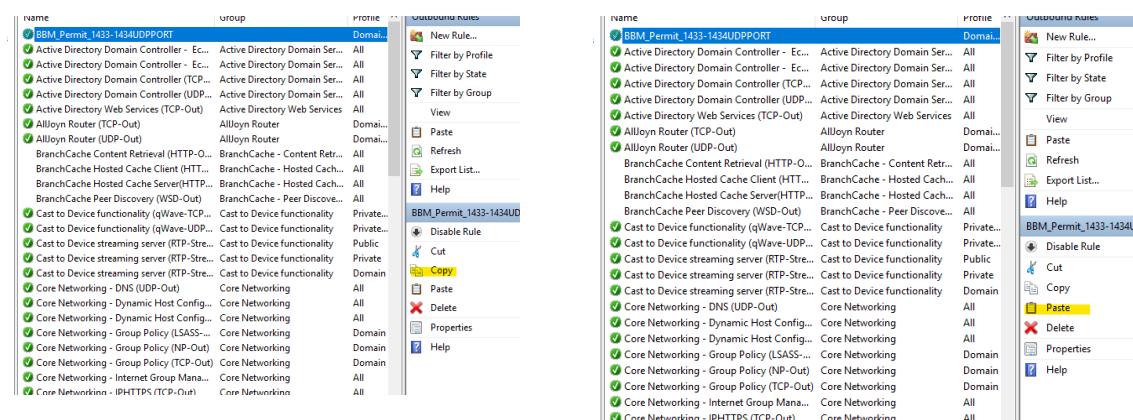


El resultado:

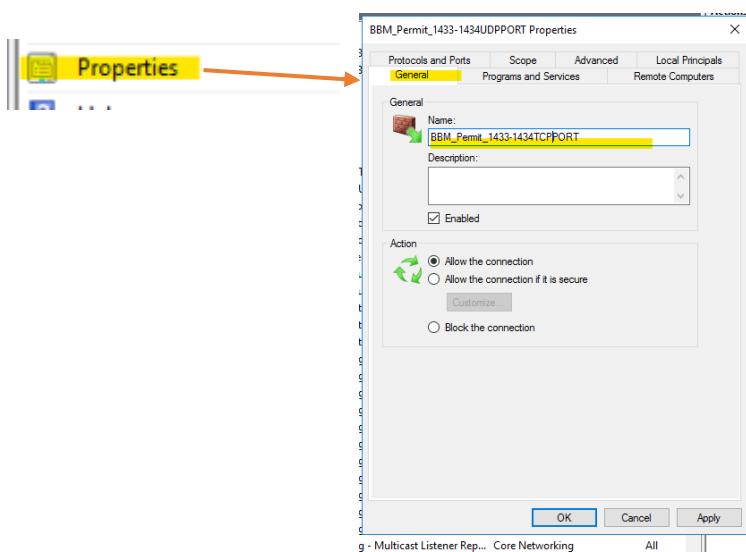


NOTA: La regla de 1433 explicada arriba la borré porque no quería duplicidades

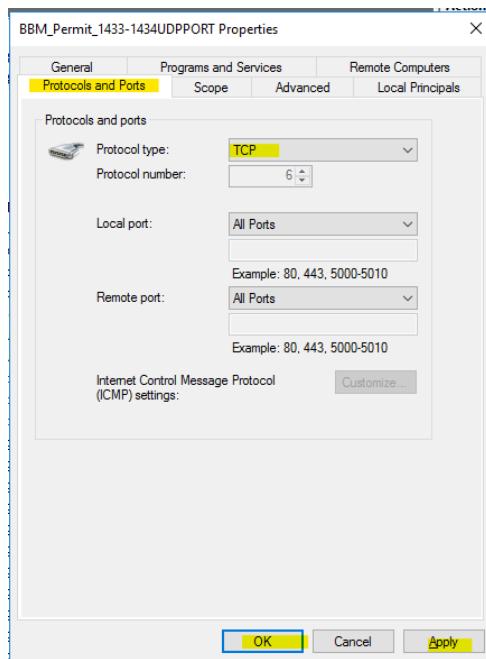
Para la regla UDP lo que hice fue copiar la regla que había creado con copy-paste:



Tocando en Properties se abre una ventana donde en la pestaña General, podemos cambiarle el nombre.

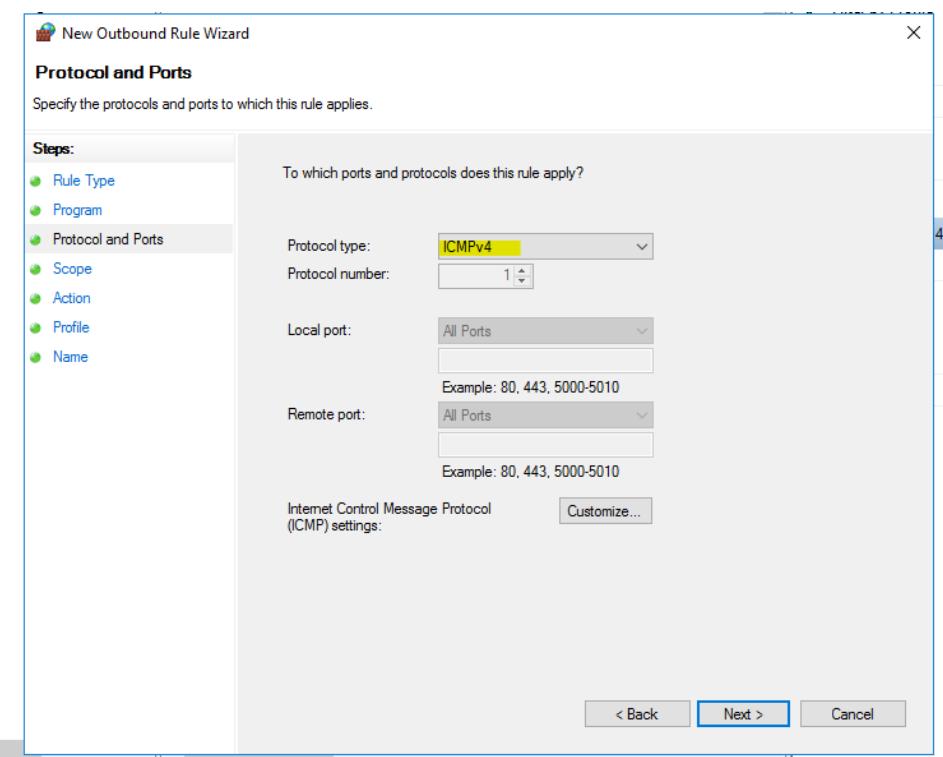


En la pestaña Protocols and Ports se cambia el puerto a TCP. Apply y OK y el cambio ya quedaría hecho

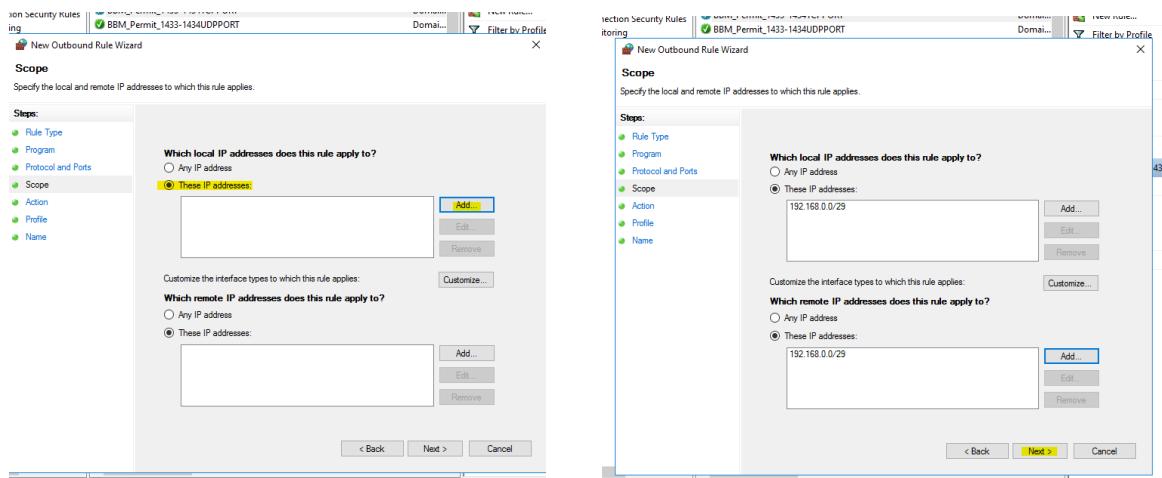


Creación de regla Inbound/Outbound para permitir la red

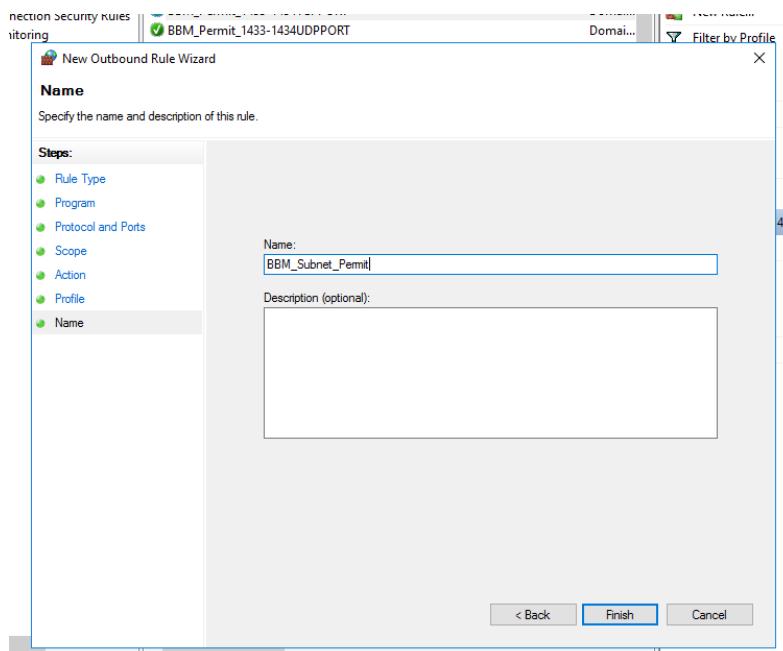
El método de creación básico ya quedó explicado, ahora dejo puestas las capturas donde cambia. En **Protocols and Ports**, elegimos de protocolo el ICMPv4, protocolo que trabaja en la capa 3 del modelo OSI y está relacionado con las IP.



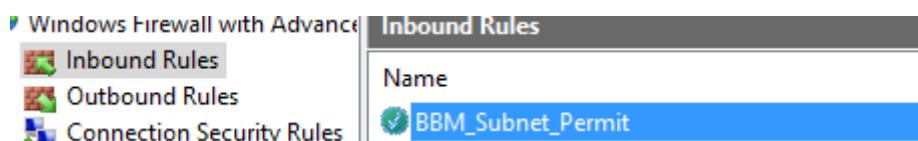
En la siguiente pestaña le decimos las IP que va a llevar. En mi caso, como hice una **subred**, agregué la subred 192.168.0.0/29.



Para finalizar, nombro la regla y acepto.



Dejo constancia de la regla finalizada:



Instalación de SSMS

SQL Server Management Studio (SSMS) es un entorno gratuito integrado para administrar cualquier infraestructura de SQL, desde *SQL Server* a *Azure SQL Database*. SSMS proporciona herramientas para configurar, supervisar y administrar instancias de SQL Server y bases de datos. Use SSMS para implementar, supervisar y actualizar los componentes de nivel de datos que usan las aplicaciones, además de compilar consultas y scripts.

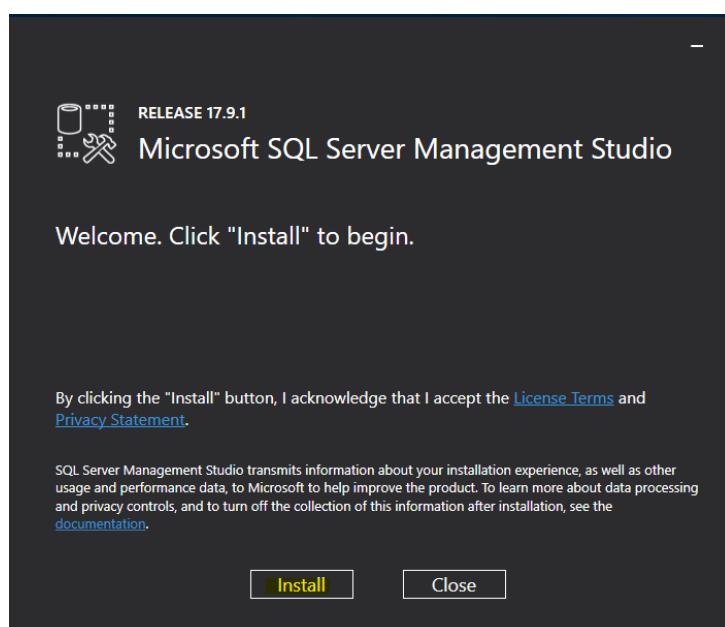
SSMS se usa para consultar, diseñar y administrar bases de datos y almacenes de datos, estén donde estén, en el equipo local o en la nube.

Para descargar el programa vamos al siguiente enlace: <https://docs.microsoft.com/es-es/sql/ssms/download-sql-server-management-studio-ssms?view=sql-server-ver15>. Nosotros la versión que queremos es la 17.9.1 por lo que debemos hacer clic en un enlace a la derecha que pone *versiones anteriores* y hacer *scroll down* hasta encontrar la versión y pinchar en el enlace:

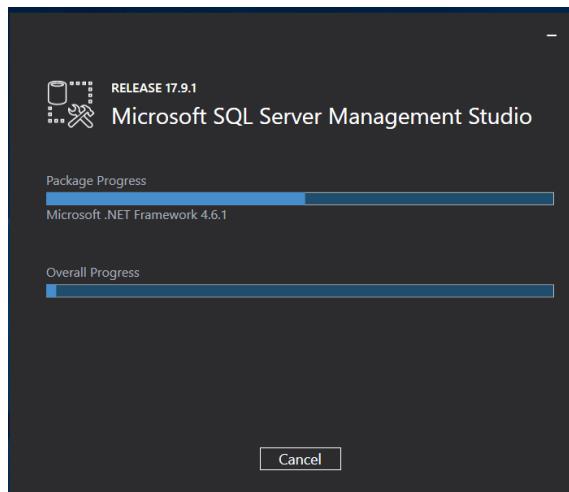
17.9.1

 [Descargar SSMS 17.9.1](#)

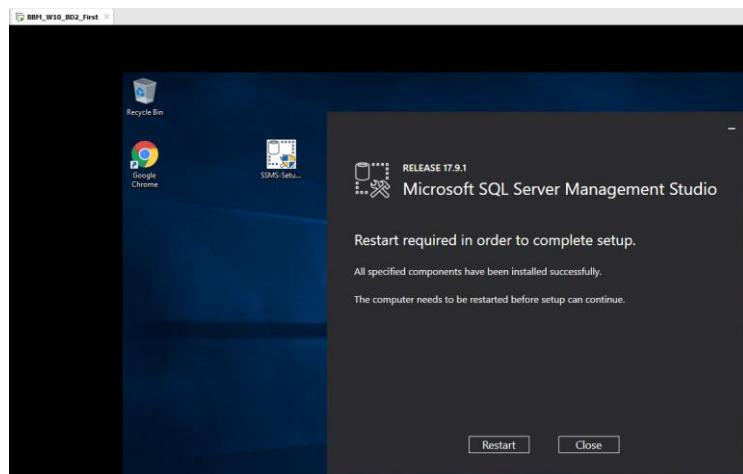
Una vez descargado, hacemos clic con el botón derecho sobre el ícono de instalación y pulsar *Run as administrator*. Permitimos al PC hacer los cambios y comienza la instalación. Se nos abre la primera ventana y hacemos clic en **Install**.



En la siguiente ventana se queda cargando procesos durante un buen período.



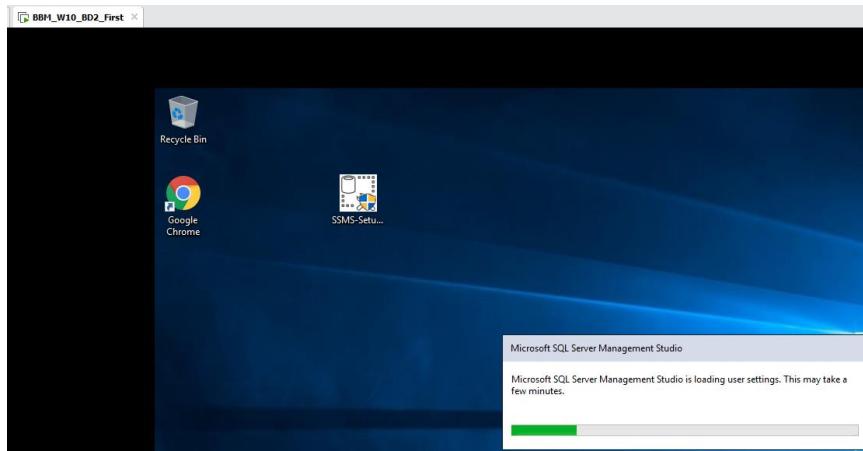
Nada más acabar de cargar, SSMS solicita un reinicio para poder completar la instalación



Una vez reiniciamos el PC, ponemos en la barra de tareas el SSMS y lo iniciamos para ver que todo está correctamente.

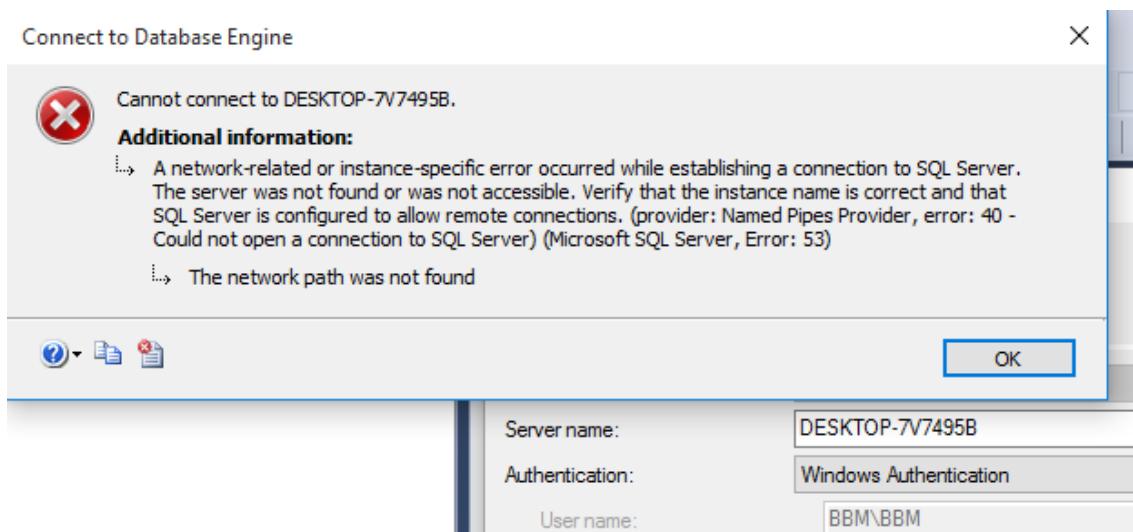


El programa se queda con una ventana de carga la primera vez comprobando todo lo instalado, y una vez hecho esto ya se inicia el programa, donde como el curso pasado ya elegiremos servido, autenticación, etc....



Problemas en el inicio de SSMS

Al instalar el SSMS me he encontrado con que el nombre del servidor no era el deseado. Tras investigar un poco, encontré que los comandos `sp_dropserver <old_name>` y `sp_addserver<new_name>` eran los necesarios para resolver el problema. Una vez ejecutado esto hay que reiniciar el servicio (desconectar y reconectar el servidor). La manera de hacerla es la siguiente botón derecho sobre el símbolo de Windows



Configuración de la red en VMWare

Como ya expliqué en la introducción, lo que vamos a hacer es interconectar tres máquinas entre sí. Para ello, he realizado un **Subneteo** de la red. Como vamos a conectar 3 **host**, la regla para ello es $2^n - 2 = n^o$ hosts, por tanto, para conseguir meter 3 hosts en la red sería $2^3 - 2 = 6$ hosts.

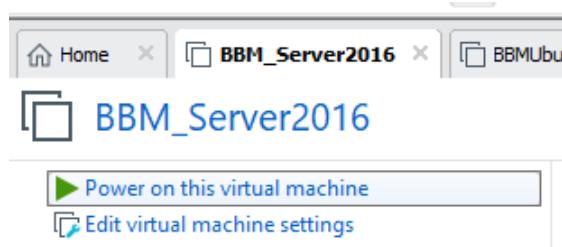
Una vez sabemos esto, cogemos una red privada de **clase C 192.168.0.0/16**. Obvio esto así nos permite 65.535 direcciones IP, con lo que vamos a reducir el número de IP según los hosts hasta dejarlo como en la siguiente tabla:

EQUIPO	IP	MÁSCARA	GATEWAY	BROADCAST
NET	192.168.0.0	255.255.255.248	192.168.0.1	192.168.0.7
W. SERVER 16	192.168.0.2	255.255.255.248	192.168.0.1	
W. CLIENT 10	192.168.0.3	255.255.255.248	192.168.0.1	
UBUNTU CLIENTE	192.168.0.4	255.255.255.248	192.168.0.1	
IP SOBRANTE	192.168.0.5	255.255.255.248	192.168.0.1	
IP SOBRANTE	192.168.0.6	255.255.255.248	192.168.0.1	NO LLEVA

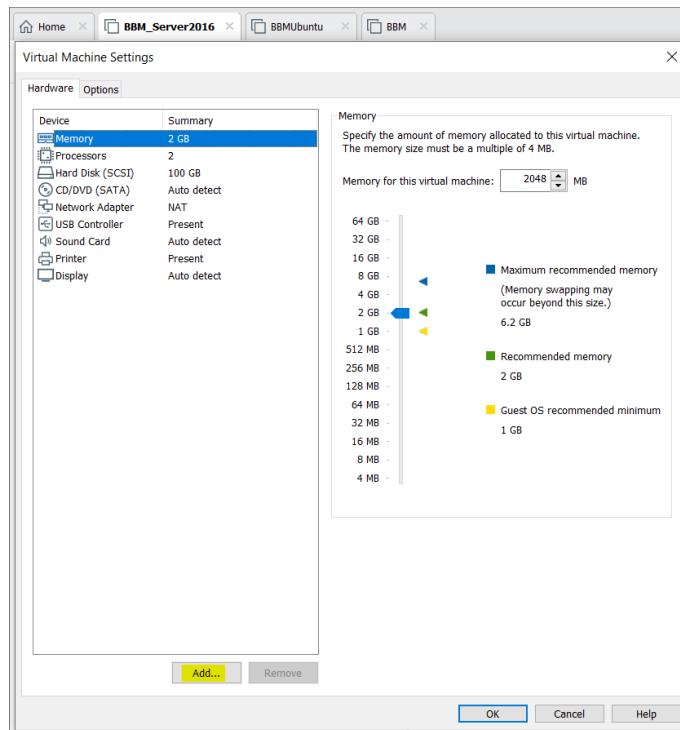
Una vez organizado qué IP va a llevar cada equipo, pasamos a la configuración de las tarjetas.

Configuración del adaptador de red en Windows Server 2016

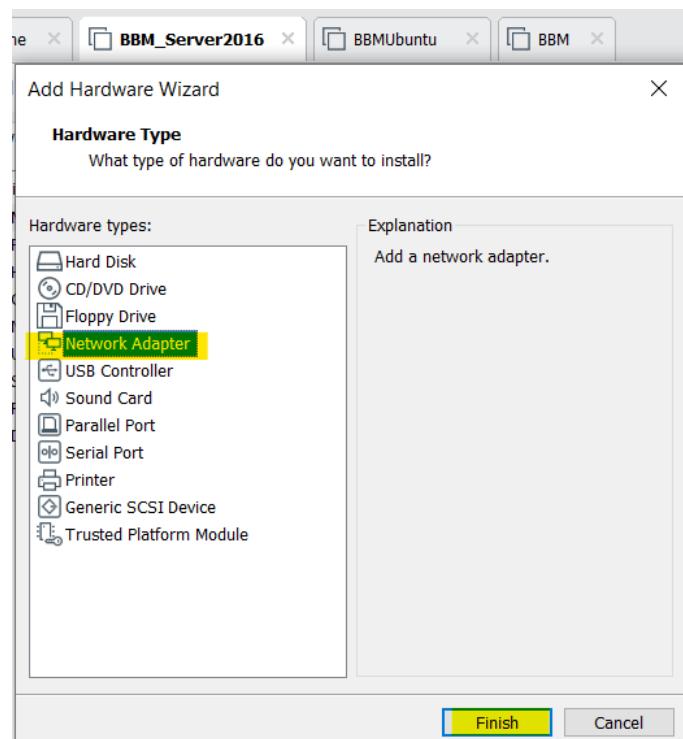
Con el equipo apagado, accedemos a las opciones de la máquina virtual, ubicada en la propia pestaña de la máquina virtual, y hacemos clic en **Edit virtual machine settings**.



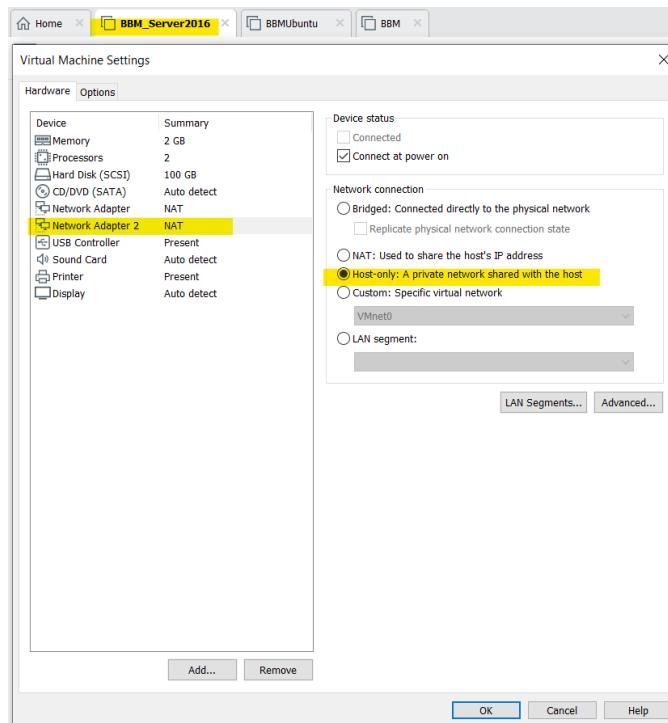
Cuando se abre la ventana para editar los componentes Hardware, seleccionamos Add..



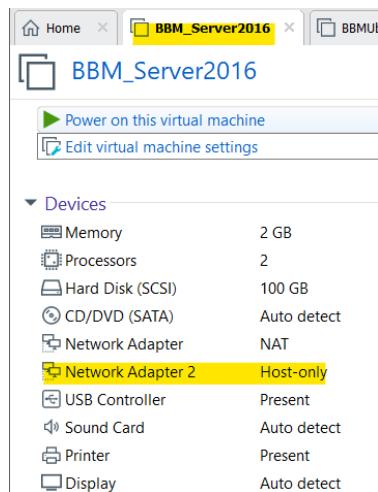
Entre las opciones a elegir tenemos **Network Adapter**. Clicamos y le damos a Finish



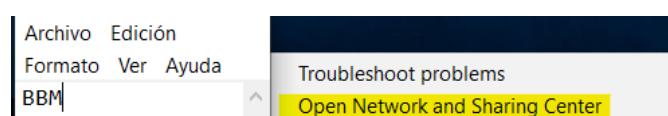
Con esta elección pasaremos a tener **dos adaptadores de red**: uno para **NAT** (el que viene por defecto) y otro **Host-only** (el que estamos instalando para conectar los tres equipos en una misma red). Seleccionamos *Host-only*.



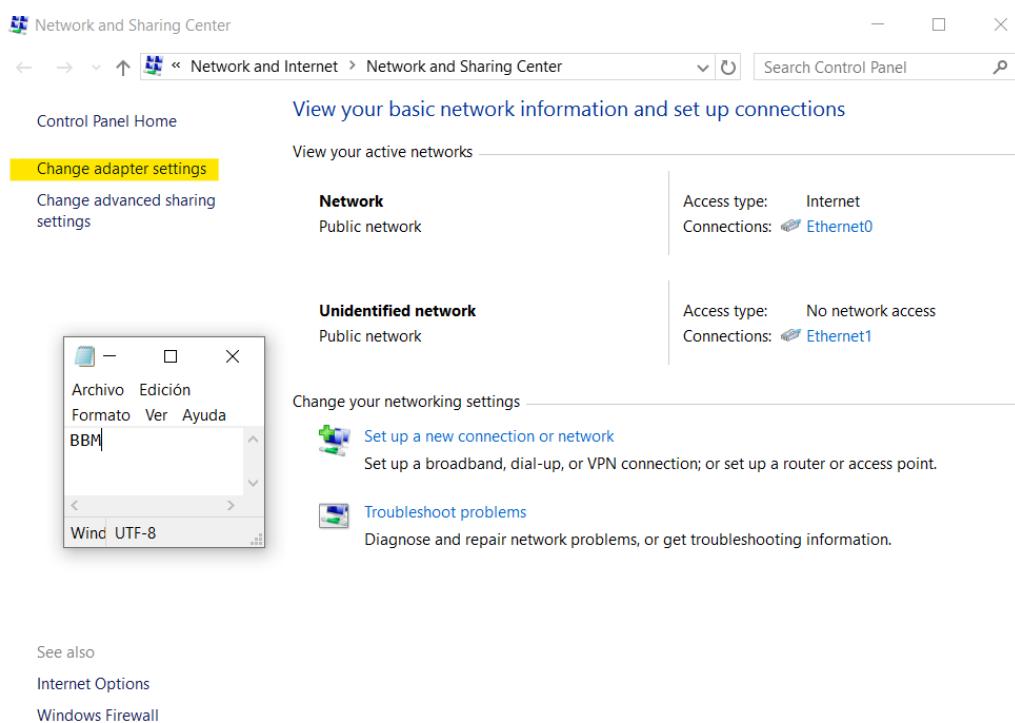
Una vez acabamos, le damos a **OK** y arrancamos el Sistema Operativo:



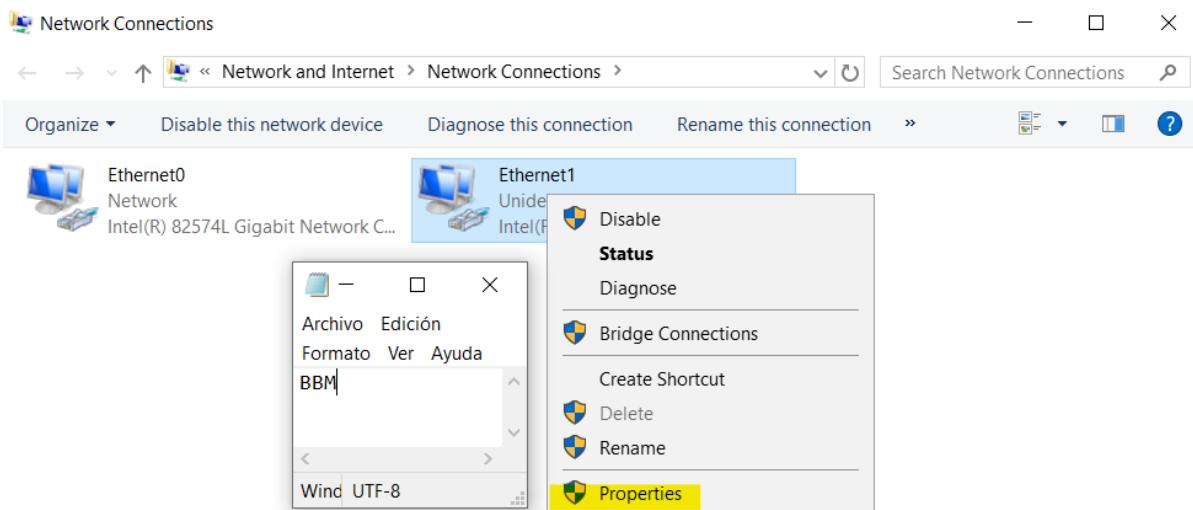
Una vez iniciado el PC, hacemos clic con el botón derecho sobre el símbolo y le damos a **Open Network and Sharing Center**.



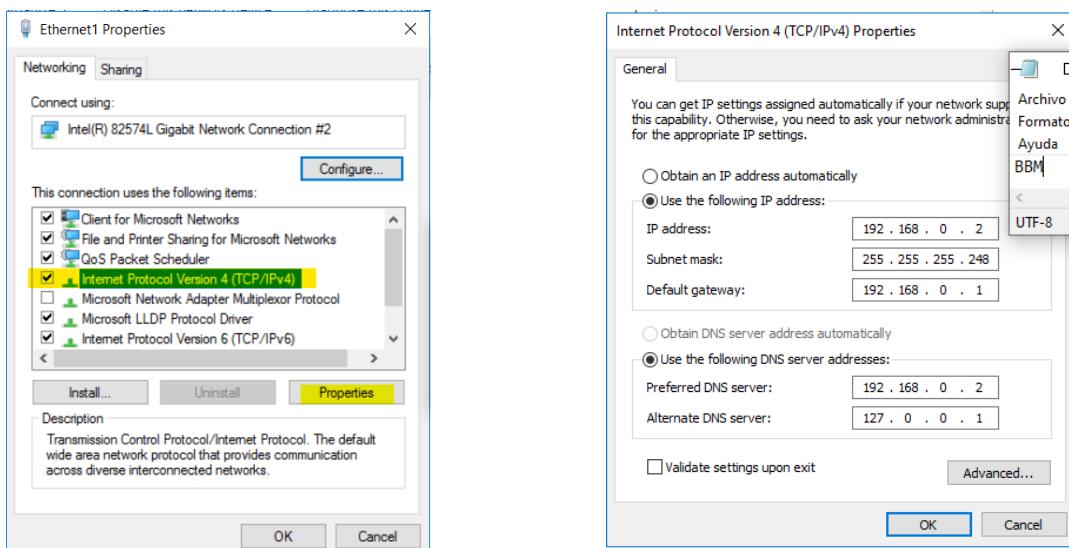
En las opciones de la izquierda, le damos a **Change Adapter Settings**.



Ahora aparecen dos cables de conexión. Generalmente **Ethernet 0** corresponde con la red que sale a Internet y **Ethernet 1** con la red local que se pretende crear. Hacemos clic con el botón derecho en la **Ethernet 1** y le damos a **Propiedades**.

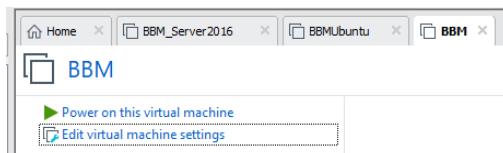


Se abre una ventana en la que podemos seleccionar varias cosas. Entre ellas, vamos al **protocolo IPv4** y le damos a **Propiedades**. En la ventana que se abre normalmente es meter los datos de la tabla que dejamos arriba. Como **DNS** metemos la IP del **Server** como **Preferred DNS**.

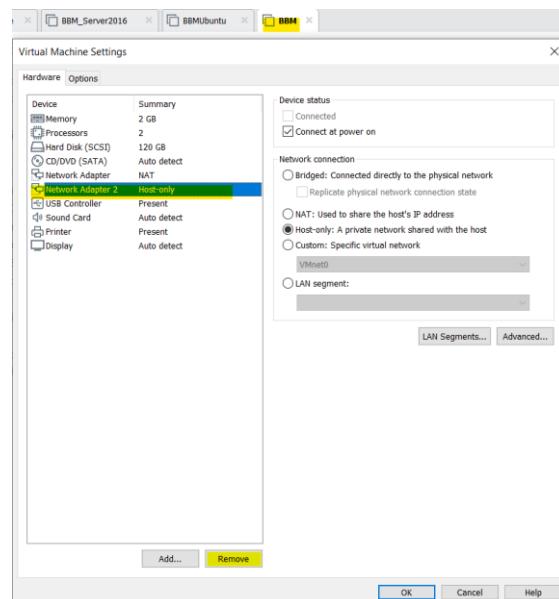


Configuración del adaptador de red en Windows 10

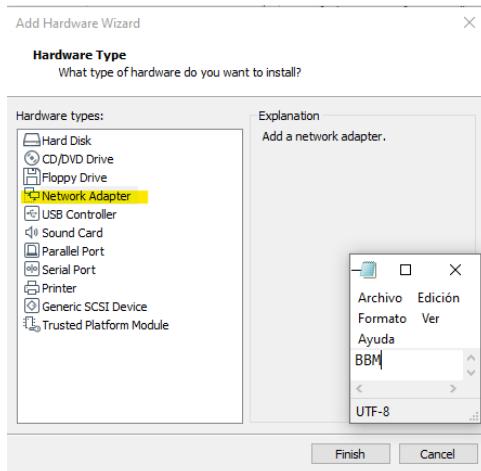
Exactamente igual que la configuración del Server. Con el equipo apagado, accedemos a las opciones de la máquina virtual, ubicada en



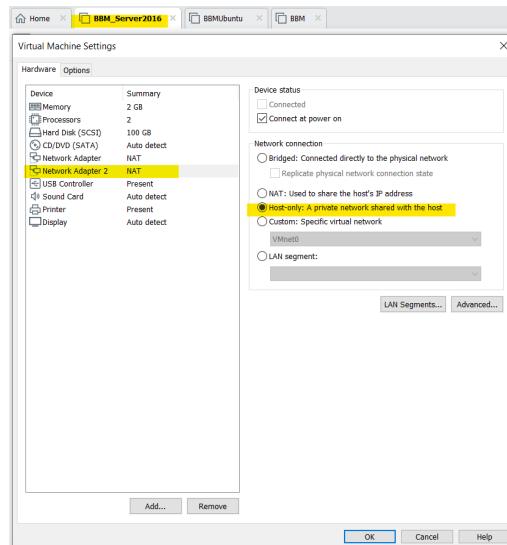
Cuando se abre la ventana para editar los componentes Hardware, seleccionamos Add..



Entre las opciones a elegir tenemos **Network Adapter**. Clicamos y le damos a **Finish**



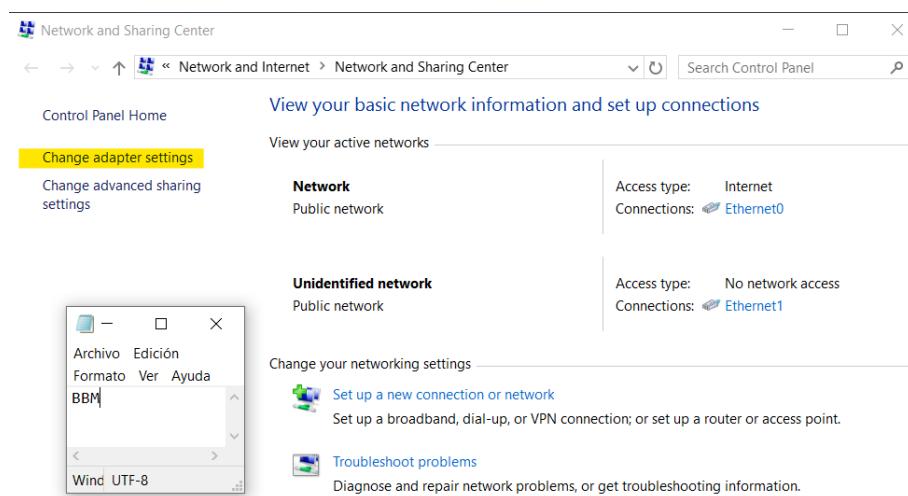
Con esta elección pasaremos a tener dos adaptadores de red: uno para **NAT** (el que viene por defecto) y otro **Host-only** (el que estamos instalando para conectar los tres equipos en una misma red). Seleccionamos **Host-only**.



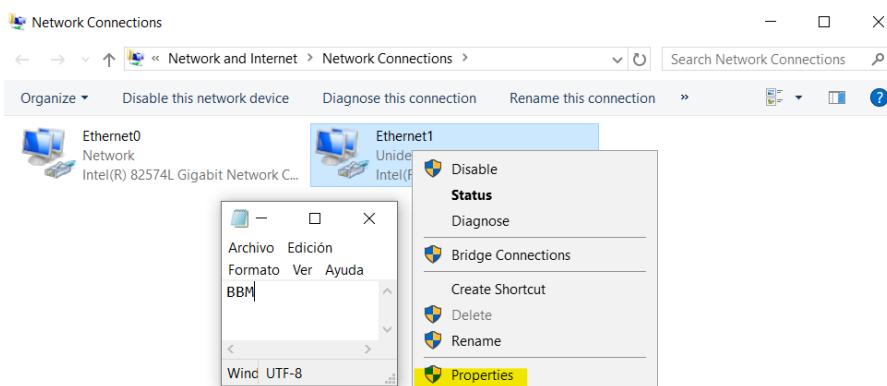
Una vez acabamos, le damos a **OK** y arrancamos el Sistema Operativo. Una vez iniciado el PC, hacemos clic con el botón derecho sobre el símbolo y le damos a **Open Network and Sharing Center**.



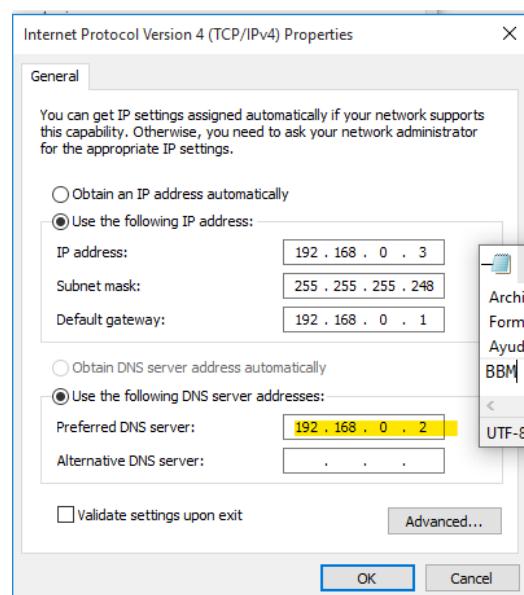
En las opciones de la izquierda, le damos a **Change Adapter Settings**.

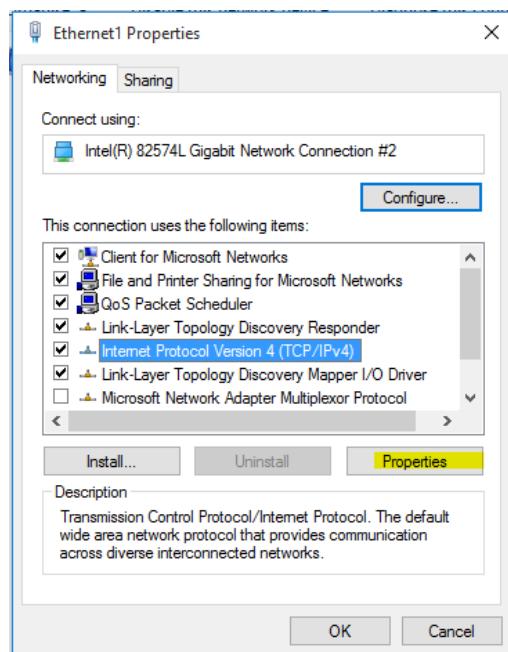


Ahora aparecen dos cables de conexión. Generalmente **Ethernet 0** corresponde con la red que sale a Internet y **Ethernet 1** con la red local que se pretende crear. Hacemos clic con el botón derecho en la **Ethernet 1** y le damos a **Propiedades**.



Se abre una ventana en la que podemos seleccionar varias cosas. Entre ellas, vamos al **protocolo IPv4** y le damos a **Propiedades**. En la ventana que se abre normalmente es meter los datos de la tabla que dejamos arriba. Como **DNS** metemos la IP del **Server** como **Preferred DNS**.





Para finalizar probamos la conexión entre ambos equipos.

```
C:\Users\BBM>ping 192.168.0.3

Pinging 192.168.0.3 with 32 bytes of data:
Reply from 192.168.0.3: bytes=32 time<1ms TTL=128
Reply from 192.168.0.3: bytes=32 time=1ms TTL=128
Reply from 192.168.0.3: bytes=32 time=1ms TTL=128
Reply from 192.168.0.3: bytes=32 time=1ms TTL=128

Ping statistics for 192.168.0.3:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Ping statistics for 192.168.0.3:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 1ms, Average = 0ms
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 1ms, Average = 0ms
```

```
C:\Users\BBM.BBMASPACE>ping 192.168.0.2

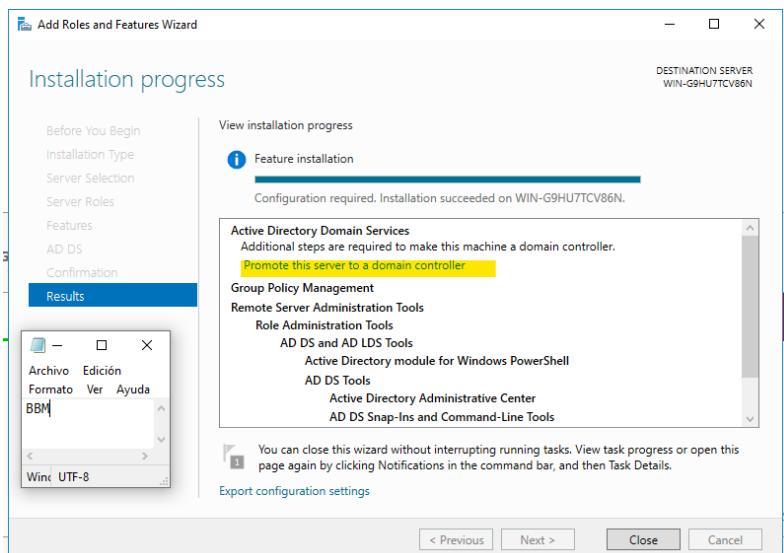
Pinging 192.168.0.2 with 32 bytes of data:
Reply from 192.168.0.2: bytes=32 time<1ms TTL=128
Reply from 192.168.0.2: bytes=32 time=1ms TTL=128
Reply from 192.168.0.2: bytes=32 time=1ms TTL=128
Reply from 192.168.0.2: bytes=32 time=1ms TTL=128

Ping statistics for 192.168.0.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 1ms, Average = 0ms
```

Configuración del Dominio

El objetivo consistirá en disponer de un controlador de dominio para administrar los recursos de nuestra infraestructura de red. Aunque aquí está así, yo primero he instalado el rol **Servicios de dominio de Active Directory**. Por ello, mi parte empieza promocionando el servidor para convertirlo en un controlador de dominio.

Después de realizar la instalación del rol *Servicios de dominio de Active Directory*, basta con hacer clic sobre el enlace **Promover este servidor a controlador de dominio** de la última pantalla del asistente. Así iniciamos la promoción.

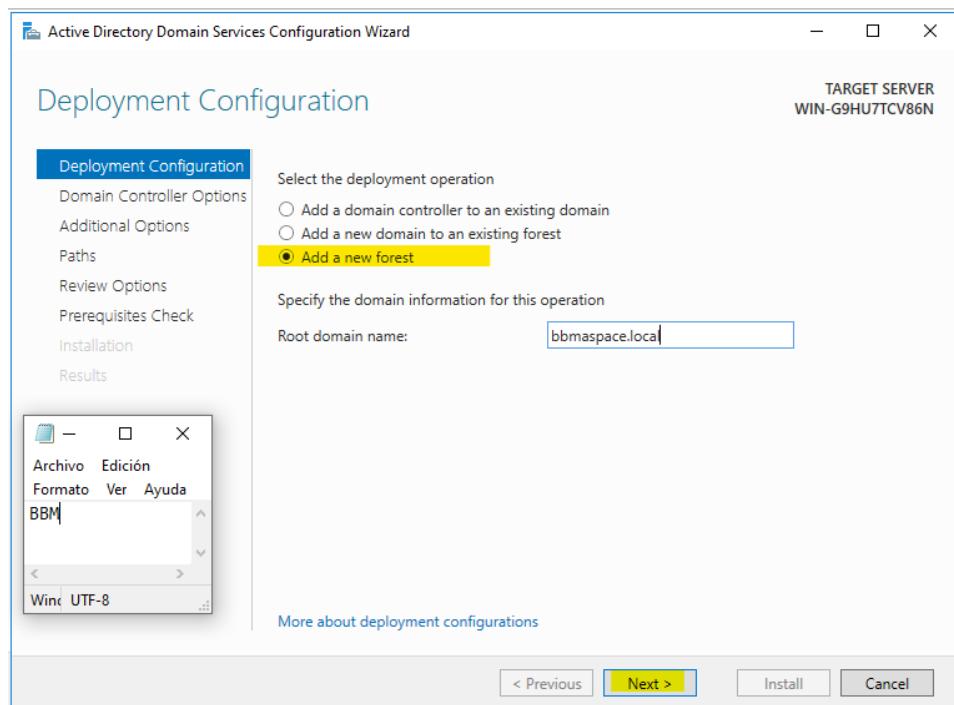


NOTA: si hubiésemos cerrado la ventana, también podremos hacer uso del icono  que aparece en la parte superior del **Administrador del servidor**.

Al hacerlo, se abrirá la ventana *del Asistente para configuración de Servicios de dominio de Active Directory*.

En la primera pantalla, deberemos indicar el tipo de operación que queremos implementar:

- Agregar un controlador de dominio a un dominio existente.
- Agregar un nuevo dominio a un bosque existente.
- Agregar un nuevo bosque.



Como en este caso estamos partiendo de una situación en la que no disponemos de infraestructura previa, la opción que deberemos elegir es la última.

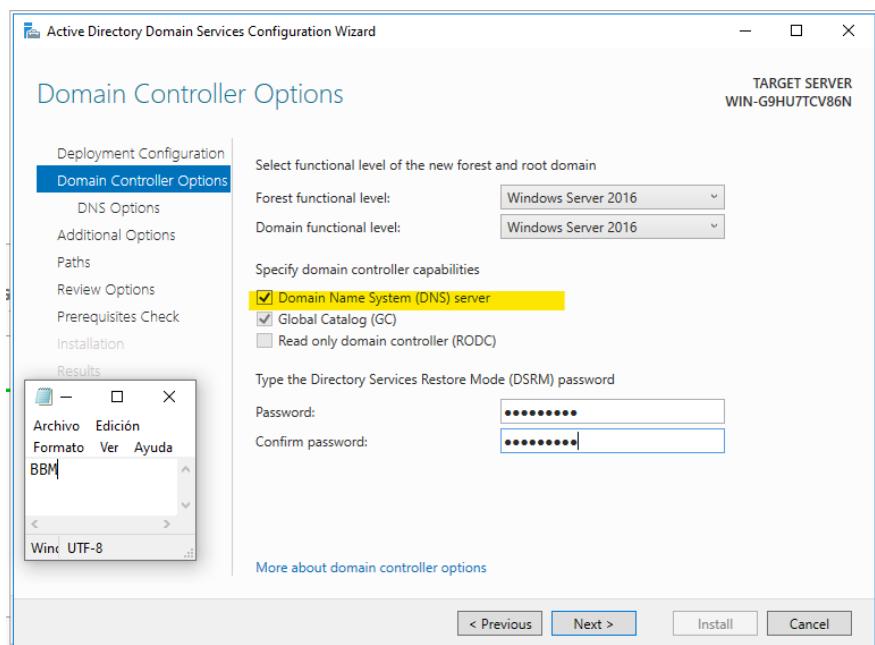
Cuando lo hagamos, en la parte inferior se nos solicitará el dominio raíz para el nuevo bosque. Ahí escribimos nuestro **nombre de dominio**. Le damos a siguiente.

NOTA: Microsoft recomienda que todos los controladores de dominio sean también servidores DNS para asegurar que Active Directory esté siempre disponible.

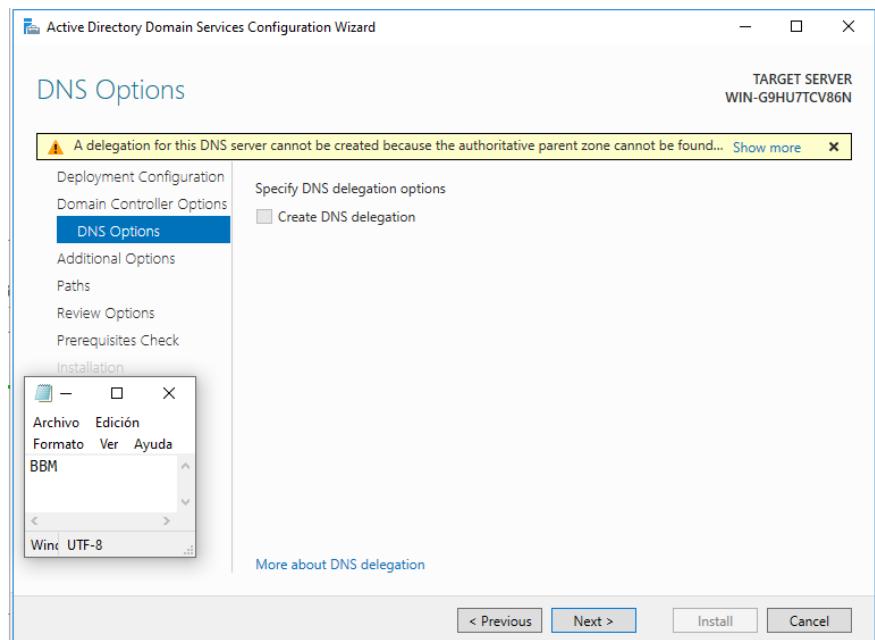
En el siguiente paso (**Opciones del controlador de dominio**) indicamos el nivel de funcionalidad del controlador. Elegimos *Windows Server 2016*.

En **Especificar capacidades del controlador de dominio**, indicaremos que el equipo también actuará como **Servidor de Sistema de nombres de Dominio (DNS)**. De las otras dos opciones más elegimos esta: **Catálogo global (GC)**.

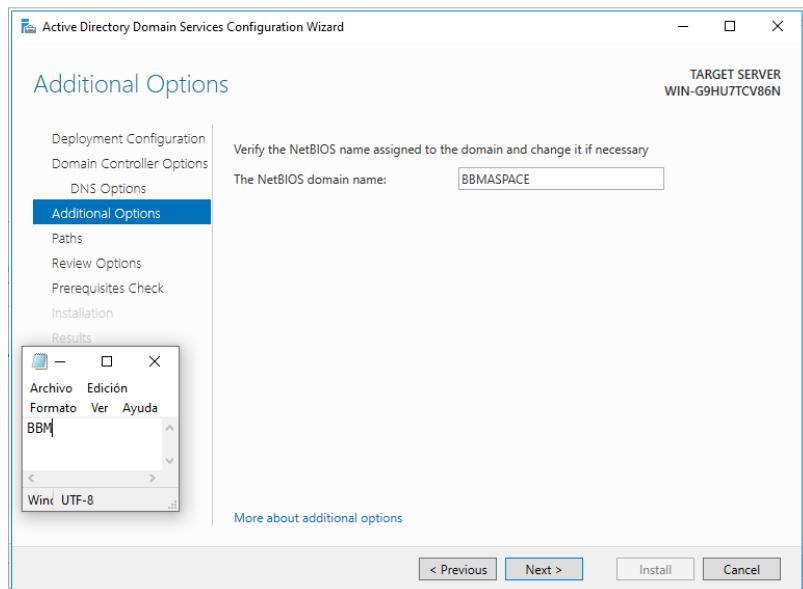
Por último, antes de cambiar de página, escribimos la contraseña del **modo de restauración de servicios de directorio (DSRM)**. Como es habitual, escribo *Abcd1234*.



En el siguiente paso, *Opciones de DNS*, como no se encuentra un Servidor DNS principal, *Windows Server* nos muestra un aviso comunicándonos que **No se puede crear una delegación para este servidor DNS porque la zona principal autoritativa no se encuentra**. Le damos a siguiente.

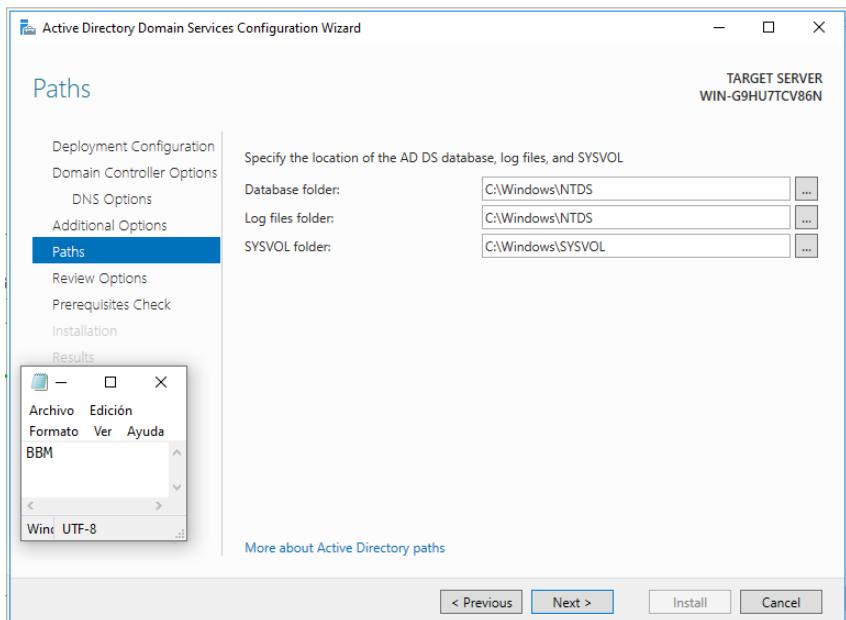


A continuación, en el paso *Opciones adicionales*, el asistente sugiere un nombre *NetBIOS* para el dominio raíz del bosque. Para estar conformado con el proyecto, de nombre puse **BBMASPACE**.

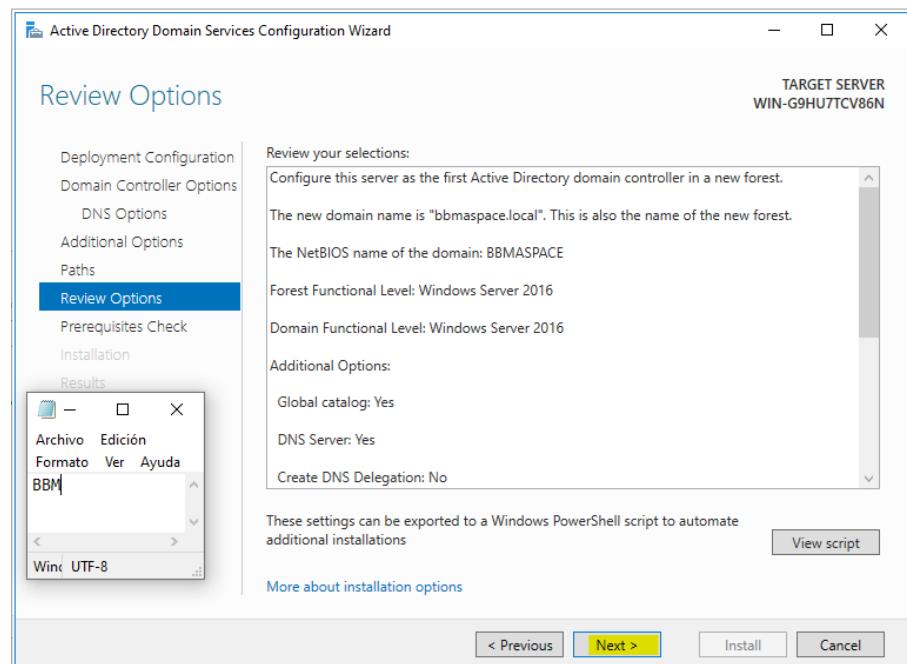


Después de esto, en el apartado *Rutas de acceso*, el asistente nos pregunta dónde queremos almacenar los archivos de trabajo de Active Directory (la base de datos, los archivos de registro y la carpeta SYSVOL).

Las dejo por defecto y le doy a siguiente.

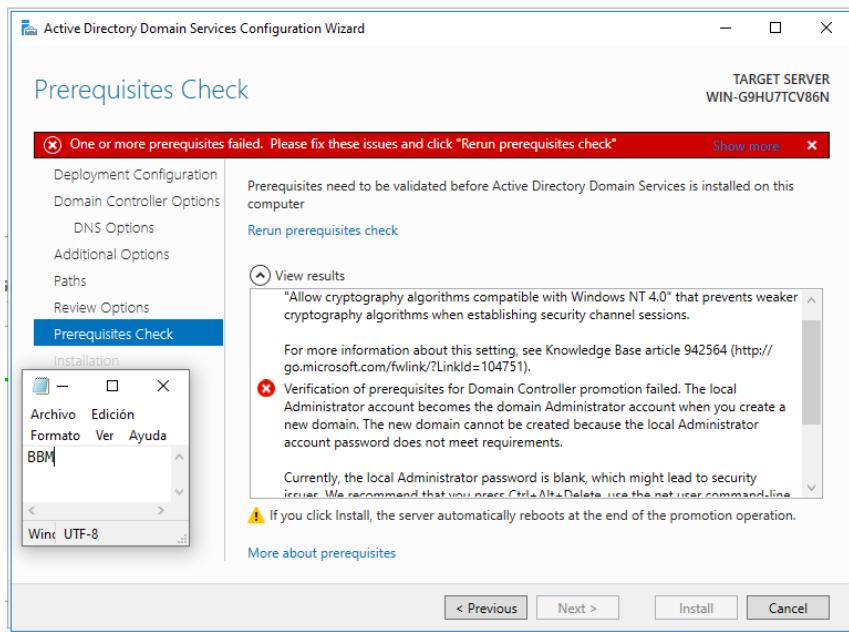


En el apartado *Revisar opciones*, el asistente muestra un resumen del proceso de instalación. Le damos a Next.

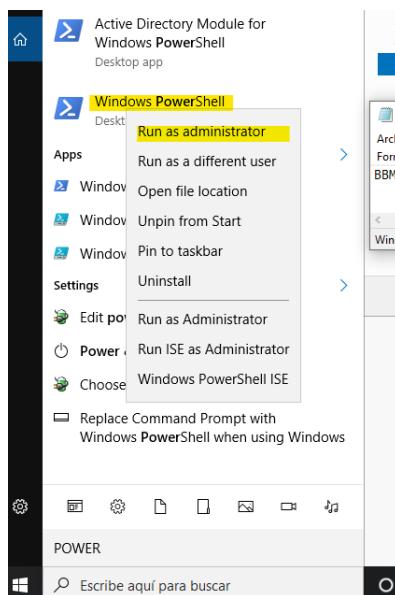


También disponemos del botón *Ver script*, que nos permite obtener un script de *PowerShell* para automatizar una instalación como ésta sin tener que volver a introducir de nuevo todos los datos.

Por último, en el apartado **Comprobación de requisitos**, se verifica que el sistema cumple todas las condiciones para convertirse en un controlador de dominio. A mí esta parte me falló:



Para arreglar este problema, voy a Cortana y entro en la Powershell como administrador:



Ejecuto el comando `net user Administrator /passwordreq:yes`

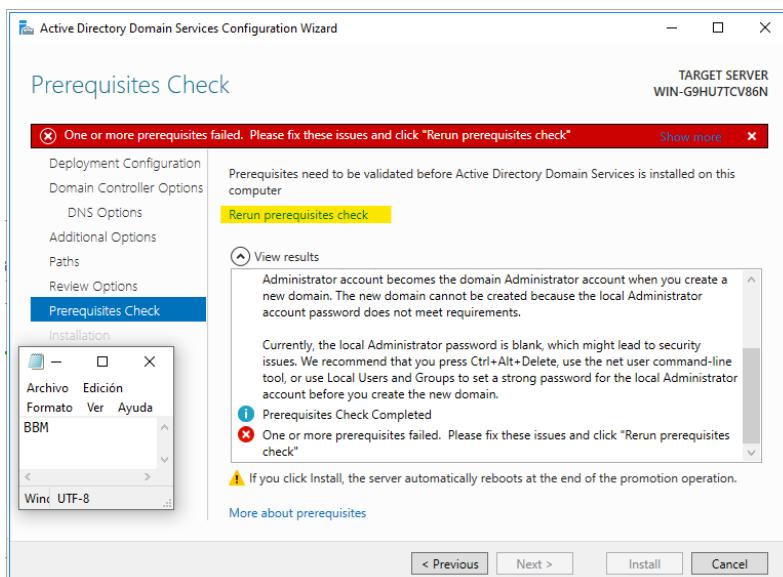
```
Administrator: Windows PowerShell
Copyright (C) 2016 Microsoft Corporation. All rights reserved.

PS C:\Windows\system32> net user Administrator /passwordreq:yes
The command completed successfully.

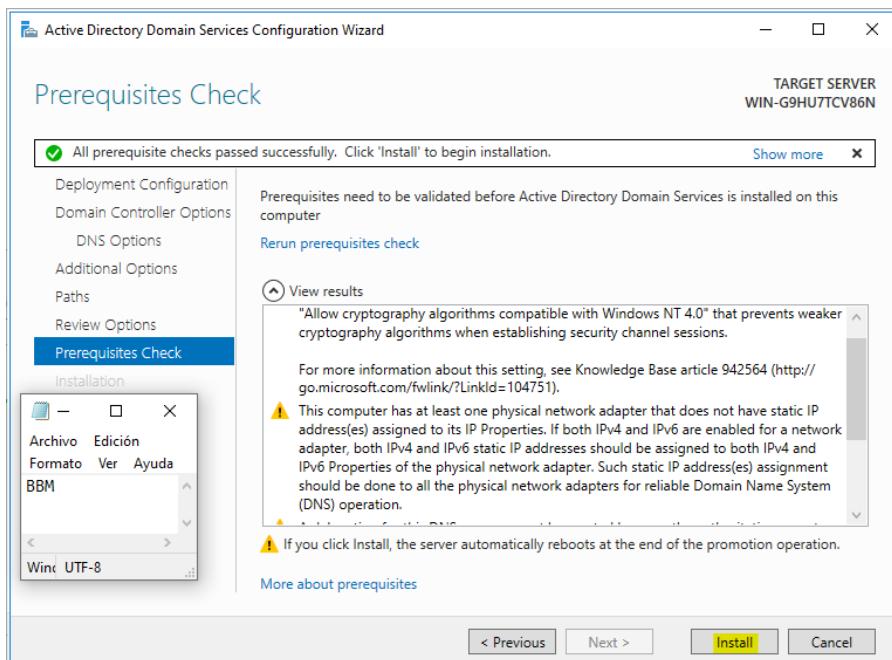
PS C:\Windows\system32>
```

A screenshot of a Windows PowerShell window running as administrator. The command `net user Administrator /passwordreq:yes` is entered and executed successfully, as indicated by the output.

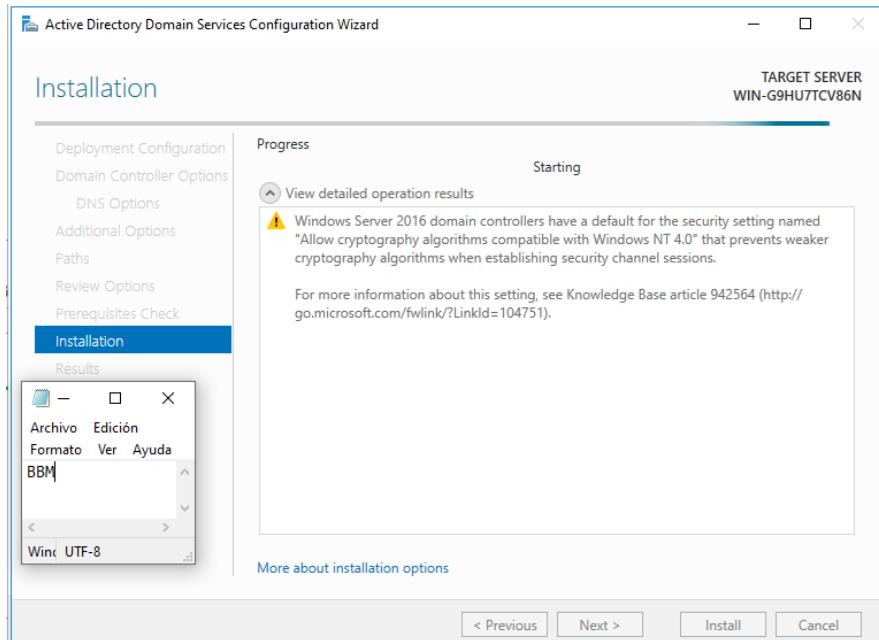
Le damos a `rerun` y ya me limpia todos los errores



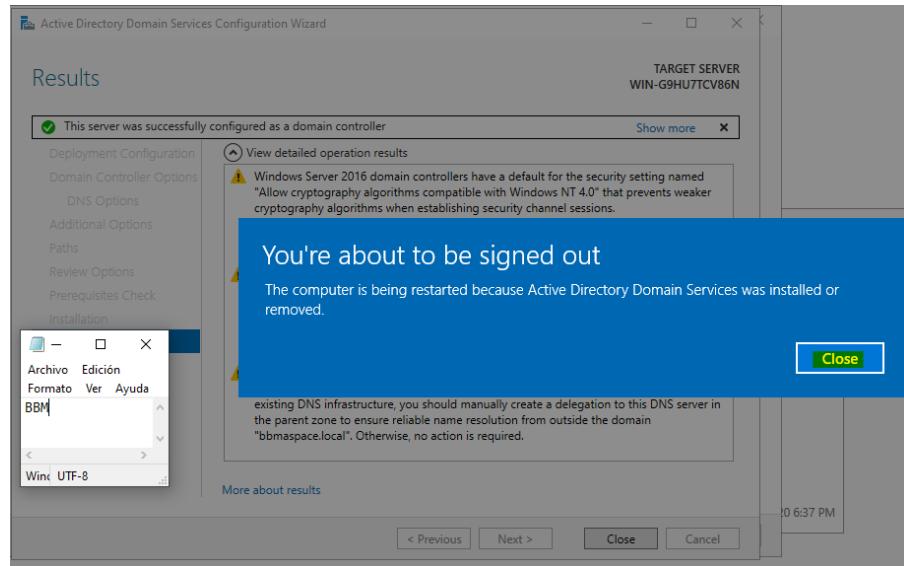
Como se ve en la siguiente imagen, ya aparece arreglado:



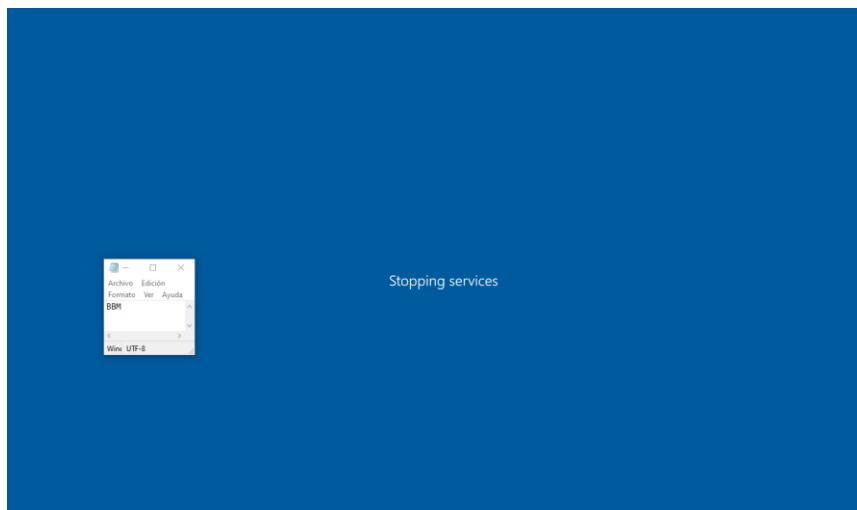
Durante el proceso de instalación seguirán apareciendo mensajes informativos que deberemos tener en cuenta para una futura configuración del servidor.



Cuando aparezca el mensaje, hacemos clic sobre el botón Cerrar.



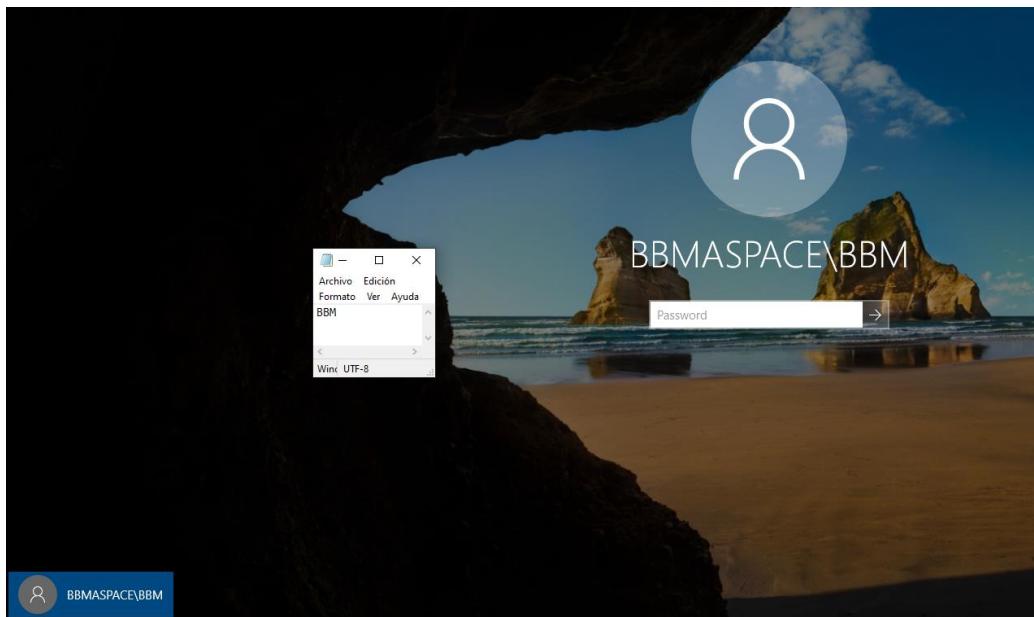
Cuando finaliza la instalación, el equipo se actualiza automáticamente y se reinicia:



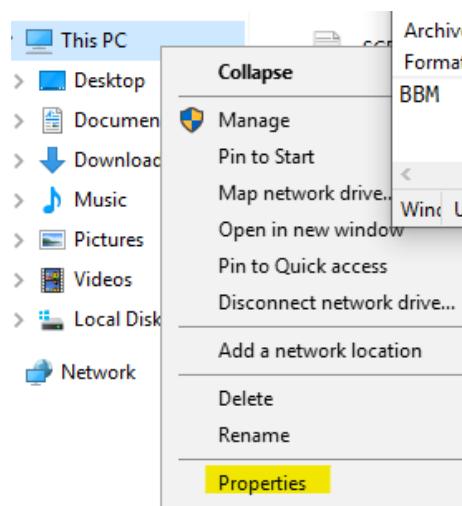
La pantalla de inicio será como siempre:



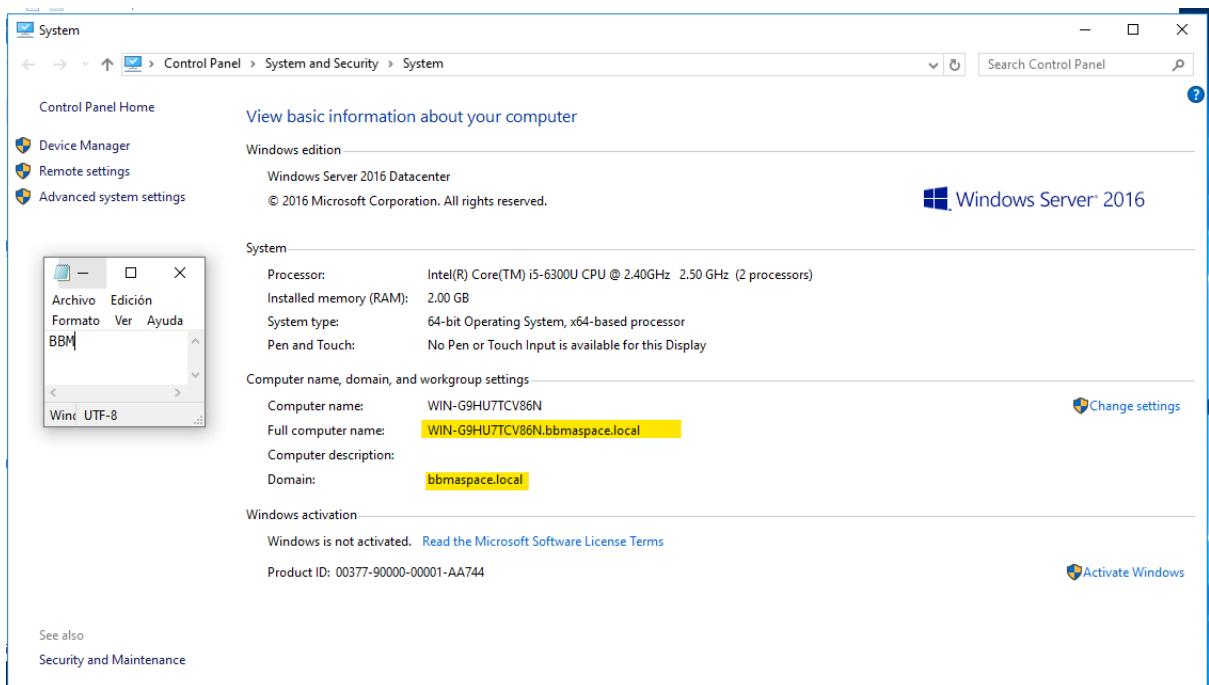
Cuando se nos solicite la contraseña de la cuenta, veremos que la cuenta aparece precedida del nombre **NetBios** (BBMASPACE en este caso) del dominio. Esta es la primera constatación de que todo ha sido correcto. Escribimos la contraseña:



Para finalizar, nos aseguramos de que todo el proceso ha sido correcto, observando la pantalla de propiedades del equipo. Vamos a **Mi PC** y hacemos clic con el botón derecho y le damos a **Propiedades**.

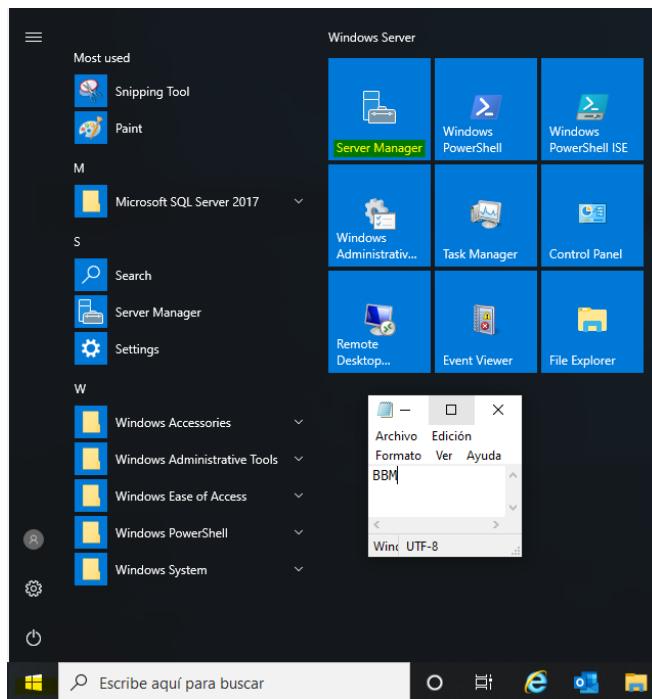


Al hacerlo, conseguimos que nos muestra la ventana **Sistema**. En ella podremos comprobar que los valores de los campos **Nombre completo** del equipo y **Dominio** son correctos.

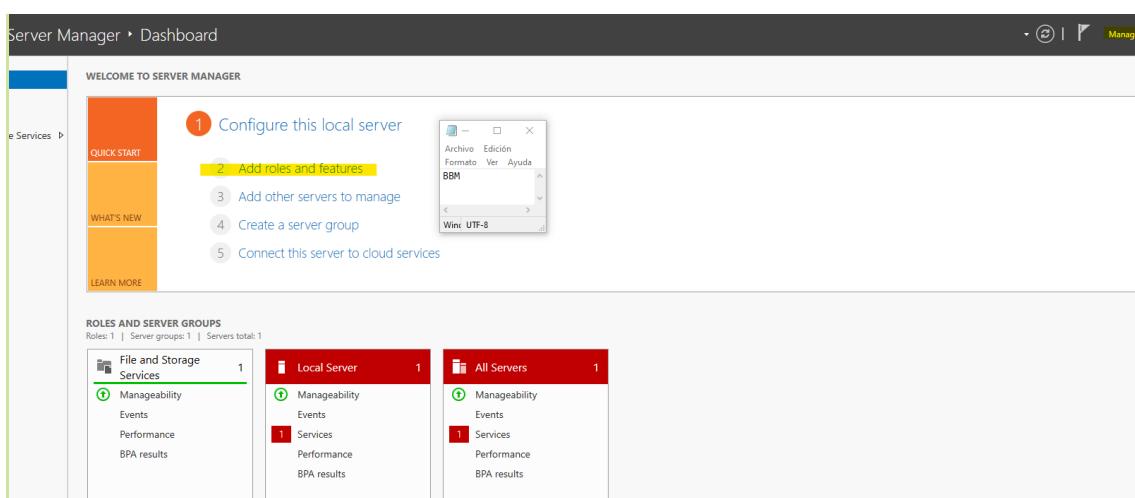


Instalar el rol Servicios de dominio de Active Directory

La instalación de roles y características de Windows Server 2016, se realiza desde la herramienta Administrador del servidor. Una vez abierta la ventana del Administrador del servidor, comenzaremos haciendo clic sobre el enlace Agregar roles y características de la página principal de la ventana.

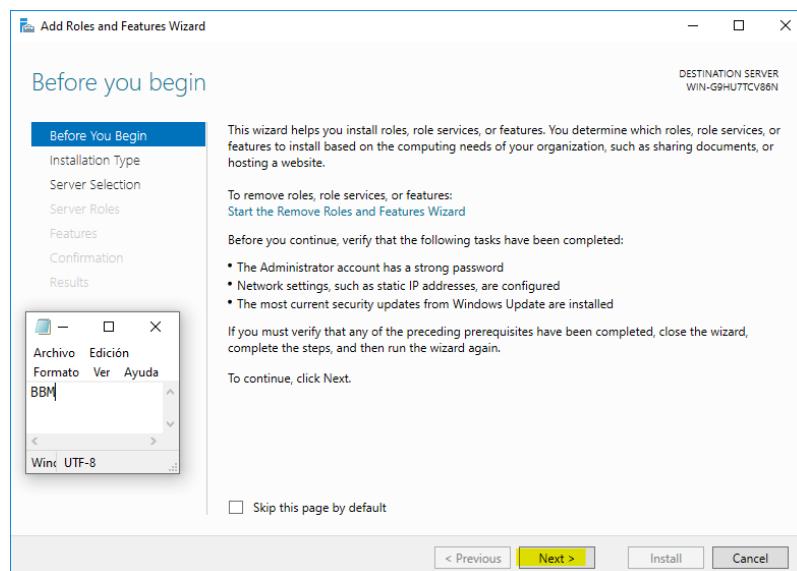


Dentro del **Server Manager**, elegimos la opción Add roles and features.

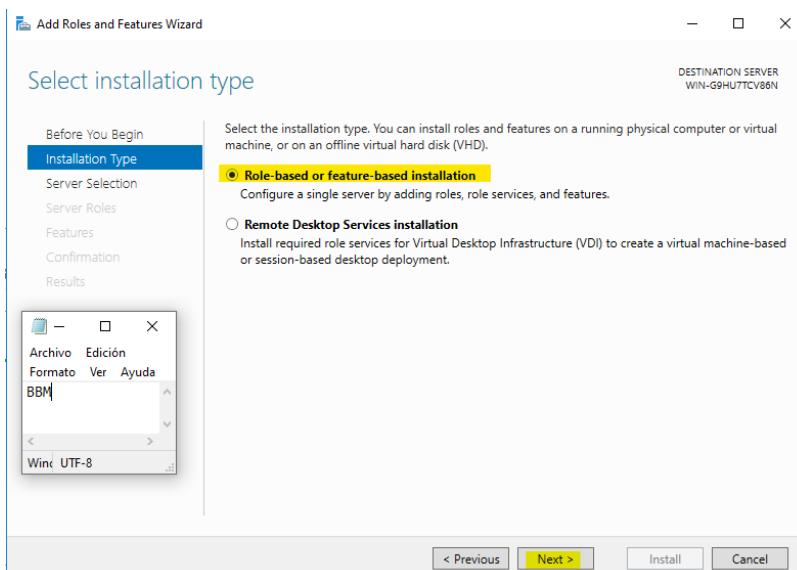


Al hacerlo, se iniciará el Asistente para agregar roles y características. Este asistente no es específico de *Active Directory*, sino que nos puede guiar a través de la instalación de otras funciones tan diversas como DNS, *Internet Information Server* (IIS), fax, etc.

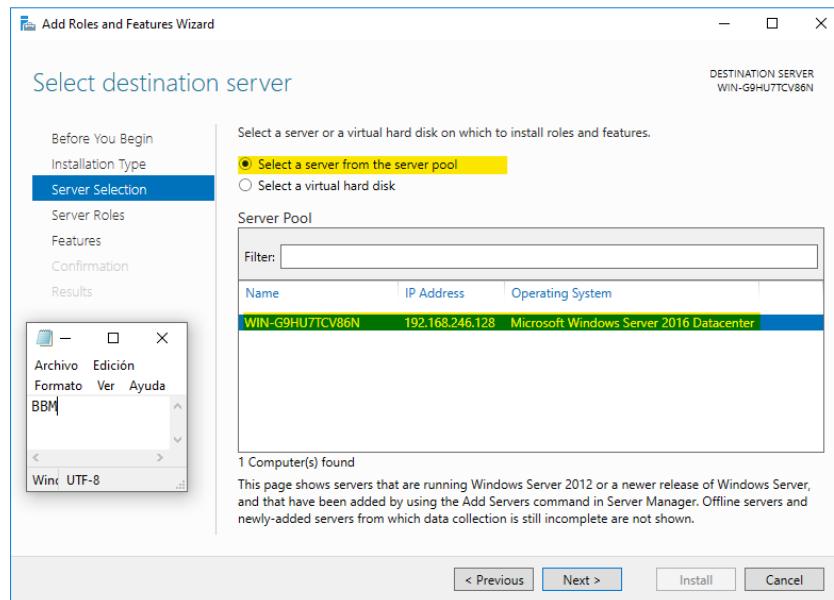
Una vez que todo sea correcto, hacemos clic en el botón Siguiente.



En la siguiente ventana elegimos **Instalación basada en características o en roles** y hacemos clic sobre el botón Siguiente.

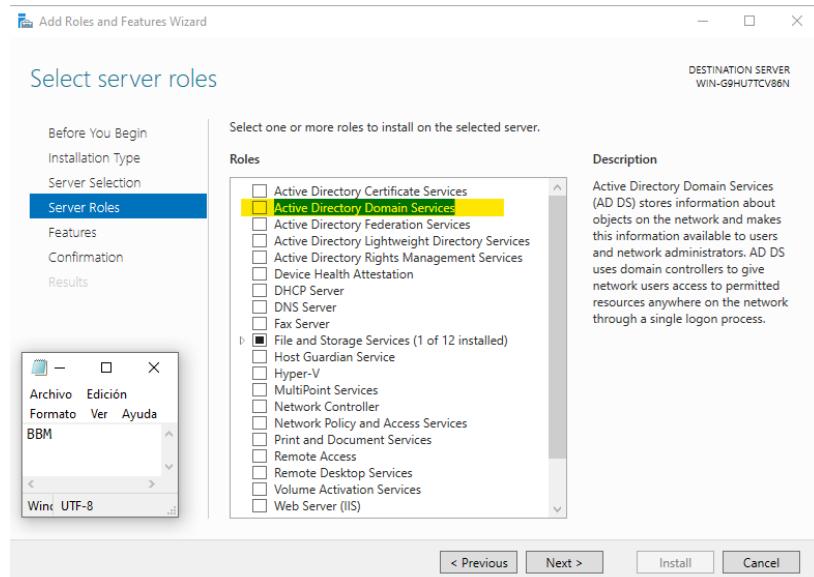


En nuestro caso, necesitamos instalar el rol del Directorio Activo en el sistema con el que estamos trabajando, por lo que la opción elegida será **Seleccionar un servidor del grupo de servidores**. De esta forma, se obtendrá una lista con los servidores de nuestra red local que ejecutan *Windows Server 2016*.

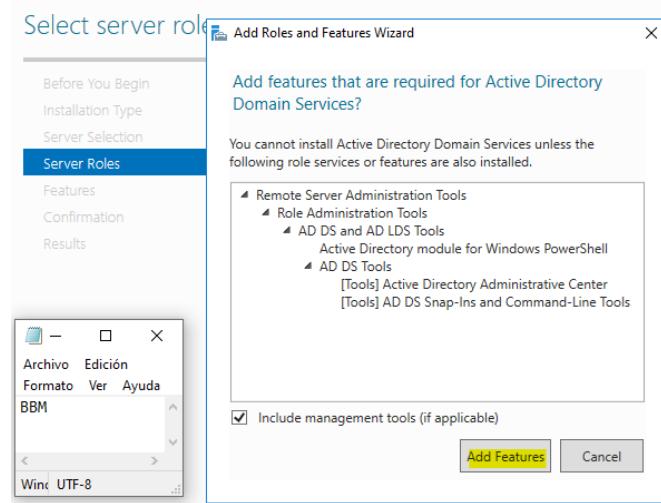


En la siguiente etapa, hay que elegir el servicio o servicios que se quieren instalar.

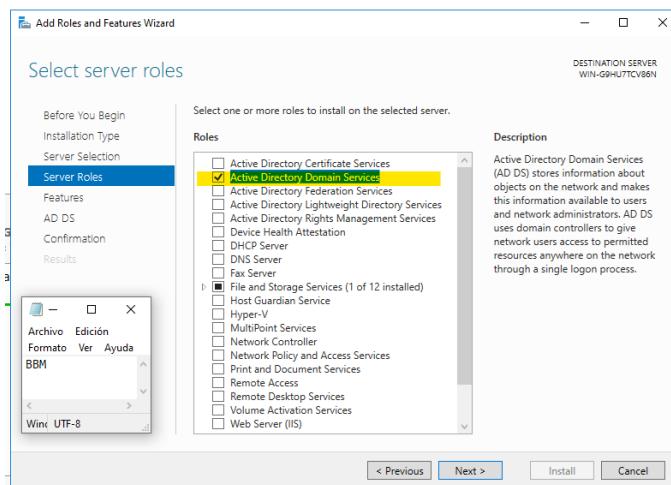
Tener en cuenta que, para que *Active Directory* funcione correctamente es preciso tener instalado un *Servidor DNS*. El asistente lo habrá seleccionado de forma predeterminada. Activamos la entrada **Servicios de dominio de Active Directory**.



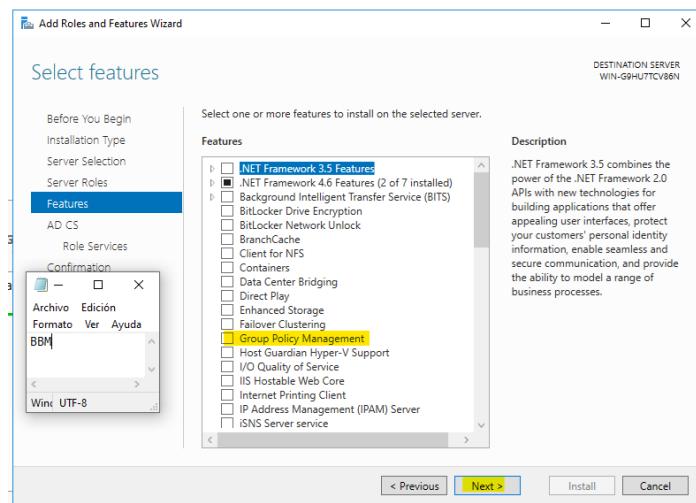
Al hacerlo, el asistente nos muestra un aviso indicando que los servicios elegidos dependen de otros roles y características que necesitaremos instalar también de forma complementaria. Le damos al botón **Agregar Características**.



Al volver a la ventana del asistente, comprobamos que la línea *Servicios de dominio de Active Directory* ya aparece seleccionada.



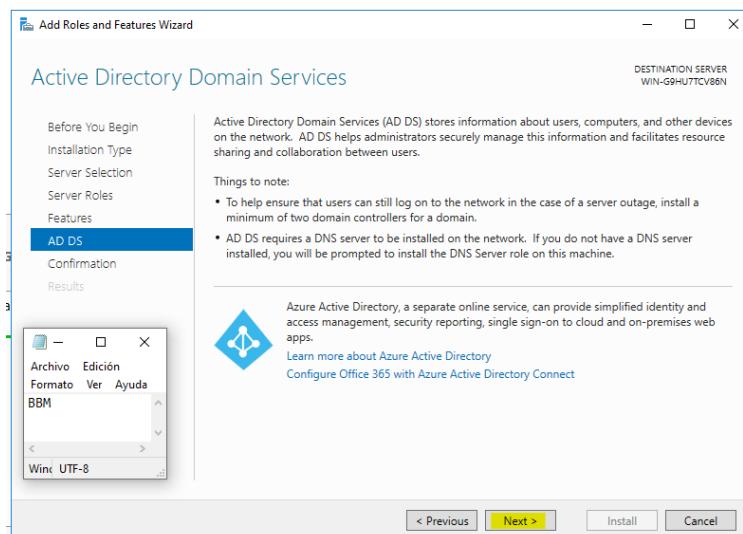
Después de seleccionar los roles, el asistente nos ofrece la posibilidad de instalar características. En *Windows Server 2016* ya se encuentra instalada de forma predeterminada. Le damos a siguiente:



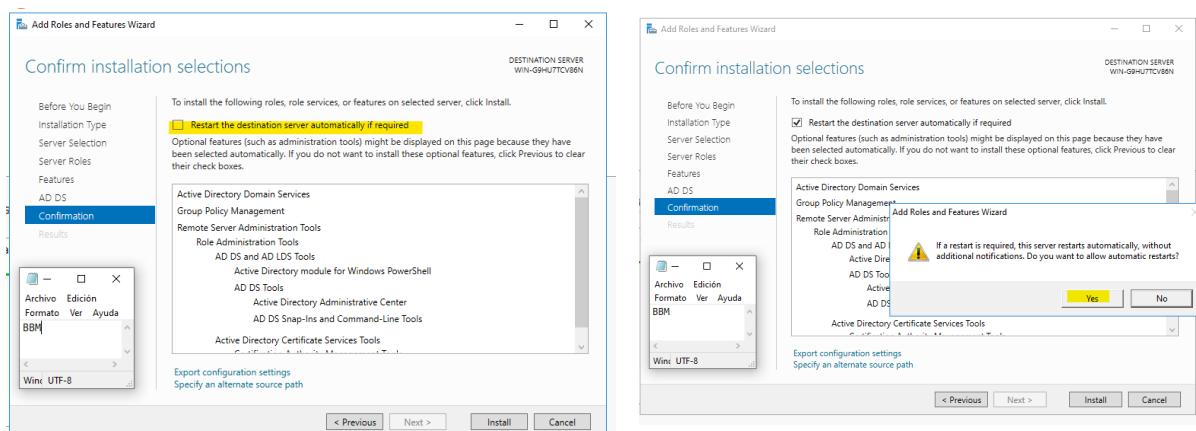
NOTA: Se puede definir el concepto de *rol* como un servicio que el equipo ofrece a los clientes de la red. Por el contrario, una *característica* es una función que usa el servidor para llevar a cabo su propia labor.

Después de esto, aparece una pantalla informativa que debemos leer atentamente. La recomendación en los foros es instalar al menos dos controladores de dominio para un determinado dominio, con el fin de aumentar la disponibilidad de la infraestructura de red. Le damos a **siguiente**.

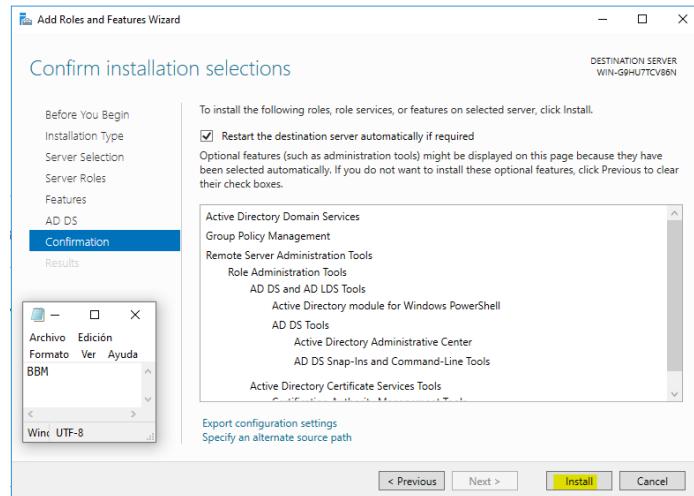
Antes de proceder con la instalación, tenemos la oportunidad de marcar la opción *Reiniciar automáticamente el servidor de destino en caso necesario*. Esto nos facilita el reinicio del sistema incluso cuando estemos administrando un equipo remoto. Recuerda que en el paso 5 podríamos haber elegido otro servidor que estuviese accesible a través de la red.



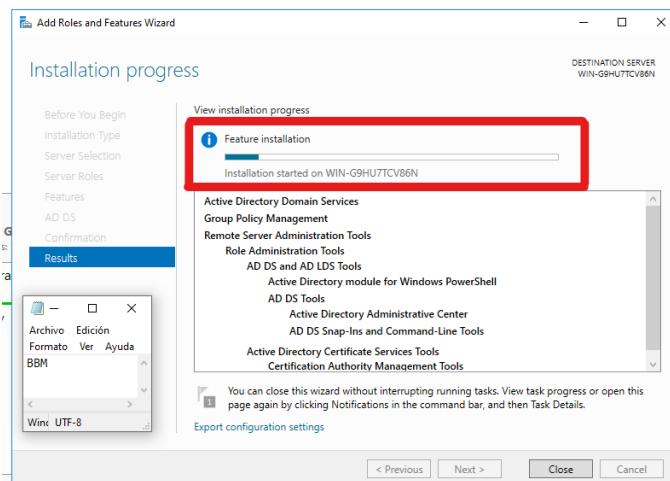
En la siguiente, tenemos un resumen, lo leemos para ver que no nos hemos equivocado. Le damos a **Install**, nos sale un *pop-up* de advertencia, la aceptamos:



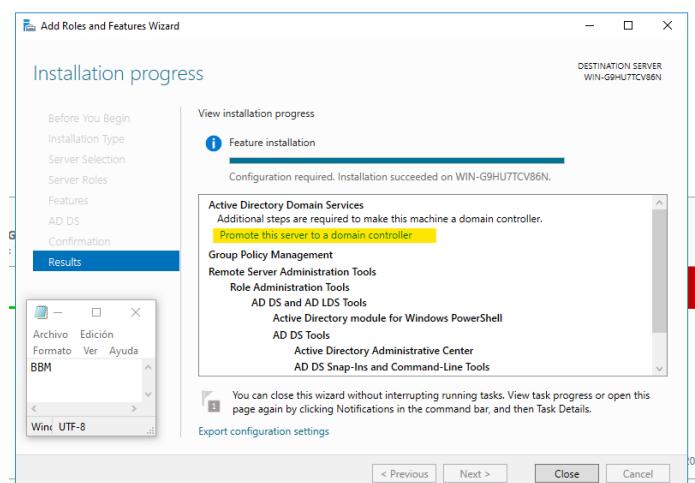
Por último, en la pantalla que resume la instalación a la que vuelve después del *pop-up*, **instalamos**:



A partir de aquí, en la parte superior de la ventana se puede ver una barra de progreso que nos mantiene informados del avance de la instalación. Esperamos a que se resuelva la carga:



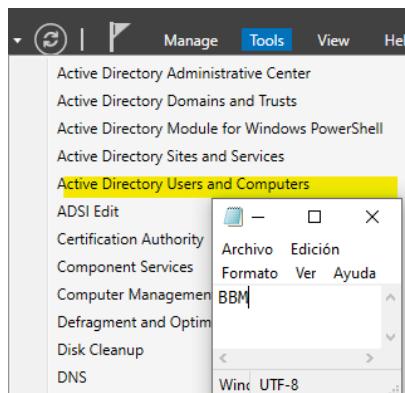
Cuando termine la instalación, aparecerá un enlace con el texto *Promover este servidor a controlador de dominio*. Hacemos clic en la frase que nos apareció.



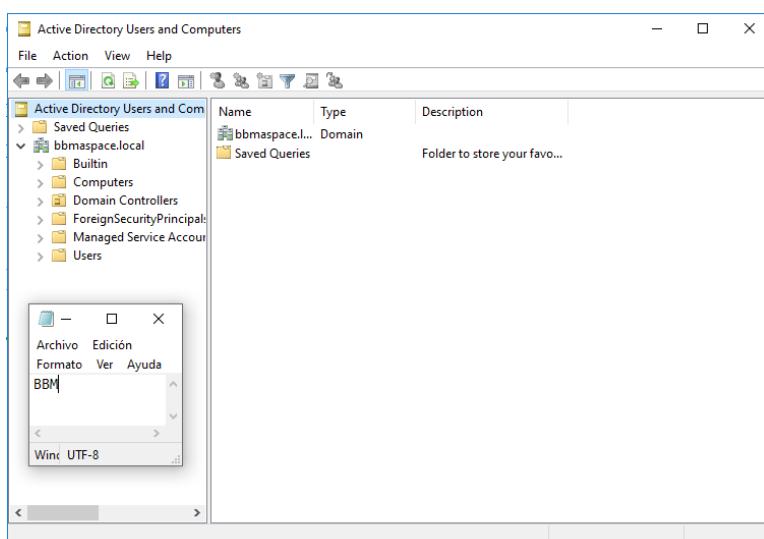
Unir la máquina con Windows 10 al Dominio

Lo último que falta del servidor es la creación de usuarios del dominio, para ello se siguen los pasos que vienen a continuación:

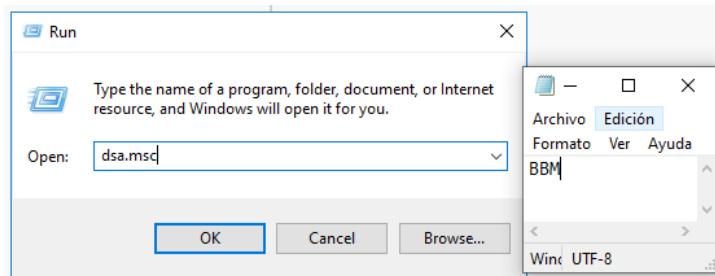
La herramienta que nos permite administrar usuarios del dominio se llama **Usuarios y equipos de Active Directory**. Para ejecutarla, sólo necesitamos acceder al menú **Herramientas del Administrador del Servidor**.



Vemos que se abre la ventana **Usuarios y equipos de Active Directory**. En ella disponemos de un panel a la izquierda donde podemos ver el dominio (**bbmaspace.local**) y, dentro del mismo, los diferentes contenedores de los que disponemos. Entre ellos, se encuentran **Builtin** y **Users**, los contenedores predeterminados para grupos y usuarios. Aquí tenemos todos los usuarios y grupos predeterminados del contenedor **Users**.

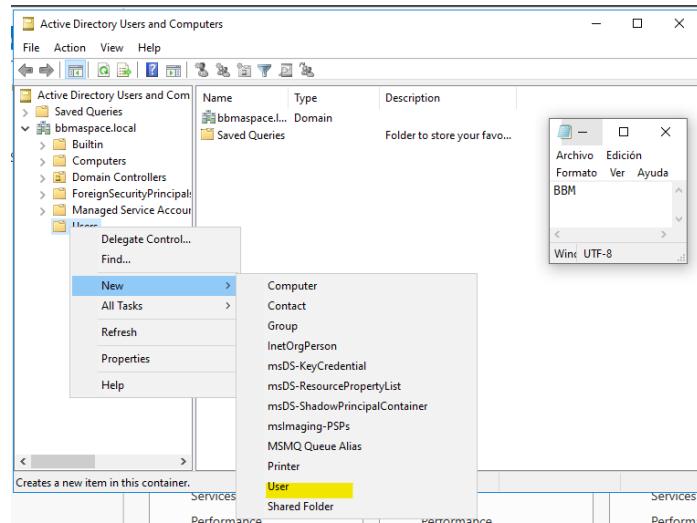


NOTA: También podemos abrir **Usuarios y equipos de Active Directory** desde la ventana Ejecutar (que puedes abrir fácilmente usando la combinación de teclas + R y escribiendo la orden `dsa.msc`).

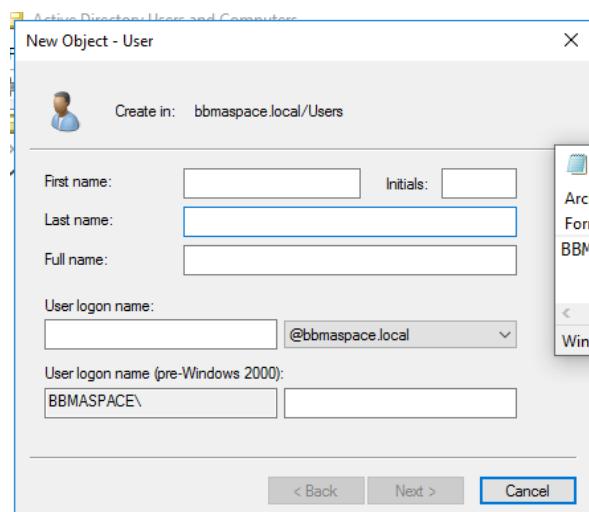


NOTA: Si tuviésemos acceso a más de un dominio, los demás también aparecerían en el panel izquierdo.

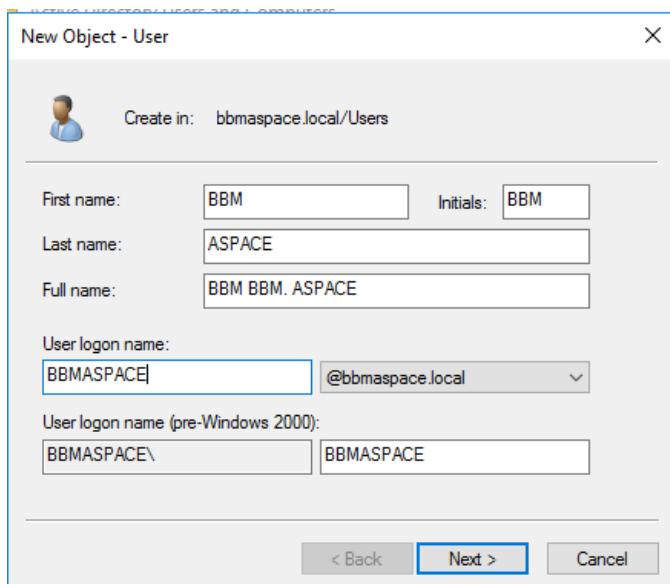
Para crear un nuevo usuario del dominio, una vez elegido el contenedor donde se almacenará el nuevo usuario, sólo tenemos que hacer clic con el botón derecho del ratón sobre él. En este caso, el contenedor será Users.



Al hacerlo, aparecerá una nueva ventana titulada *Nuevo objeto: Usuario*. Es una especie de asistente de creación de usuarios.



En el primer paso, tendremos que llenar los datos del usuario: su **Nombre, Apellidos e Iniciales**. Con ellos se formará el campo **Nombre completo** (que también se puede editar).



A continuación, escribiremos el **Nombre de inicio de sesión de usuario** que, en realidad, se compone de dos partes: el nombre propiamente dicho y el sufijo, que se elige de una lista desplegable. El campo **Nombre de inicio de sesión de usuario (anterior a Windows 2000)** tiene como objeto permitir que se conecten al dominio clientes que ejecuten Windows 95, Windows 98 o Windows NT. Al acabar clicamos en **Siguiente**.

A continuación, tendremos que escribir una contraseña para el usuario, que deberá cumplir con los requerimientos de seguridad del sistema operativo. Es decir, que de forma predeterminada deberá tener un mínimo de seis caracteres de larga y contener caracteres de, al menos, tres de los cuatro conjuntos siguientes:

- Mayúsculas del alfabeto inglés.
- Minúsculas del alfabeto inglés.
- Dígitos decimales (del 0 al 9).
- Caracteres no alfanuméricos.

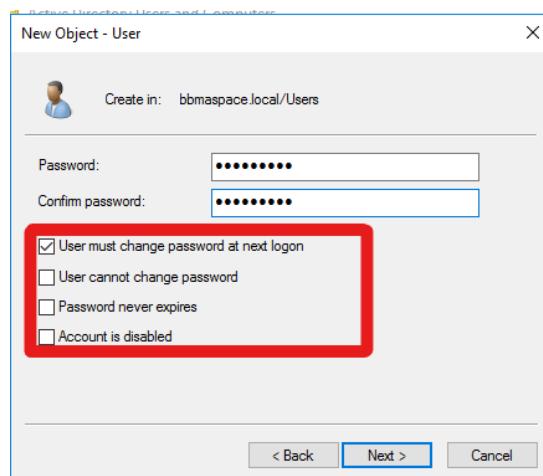
Además, en la parte inferior de la ventana disponemos de cuatro opciones:

- **El usuario debe cambiar la contraseña en el siguiente inicio de sesión:** Si lo marcamos (opción por defecto), obligaremos al usuario a cambiar la contraseña que estamos escribiendo la próxima vez que inicie sesión en el dominio. De esta forma, el usuario estará seguro de que nadie más conoce su contraseña.
- **El usuario no puede cambiar la contraseña:** Al contrario que la anterior, esta opción impide que el usuario pueda cambiar su

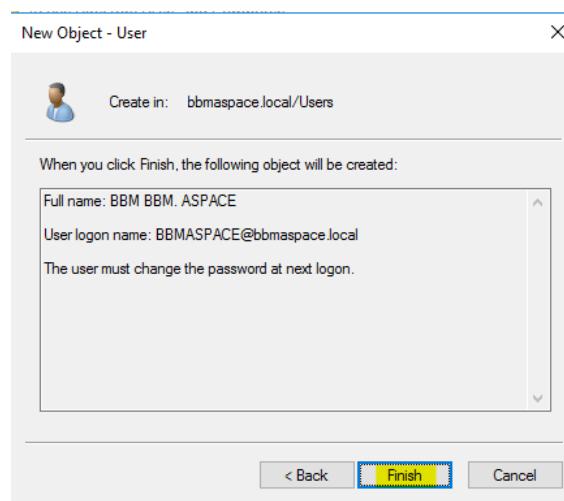
contraseña en ningún momento. Esta opción puede resultar interesante para que el administrador mantenga el control total sobre alguna cuenta temporal o de invitado.

- **La contraseña nunca expira:** hace que la contraseña no expire en el plazo que establezca el sistema operativo. Microsoft recomienda que las cuentas de servicio tengan esta opción habilitada y usen contraseñas seguras.
- **La cuenta está deshabilitada:** Mientras esta opción esté habilitada, el usuario no podrá iniciar sesión en el sistema.

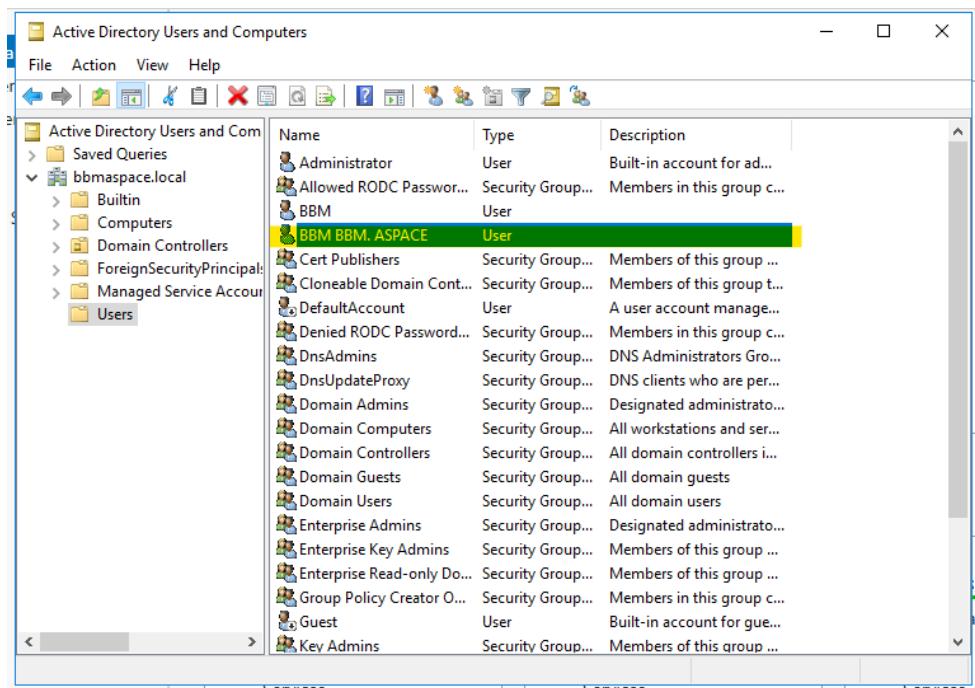
Cuando hayamos completado los datos necesarios en este paso, haremos clic sobre el botón Siguiente.



En la última ventana se nos muestra un resumen de todo lo que configuramos. Le damos a **Finish**.



Comprobamos que el usuario (o usuarios) que creamos están:



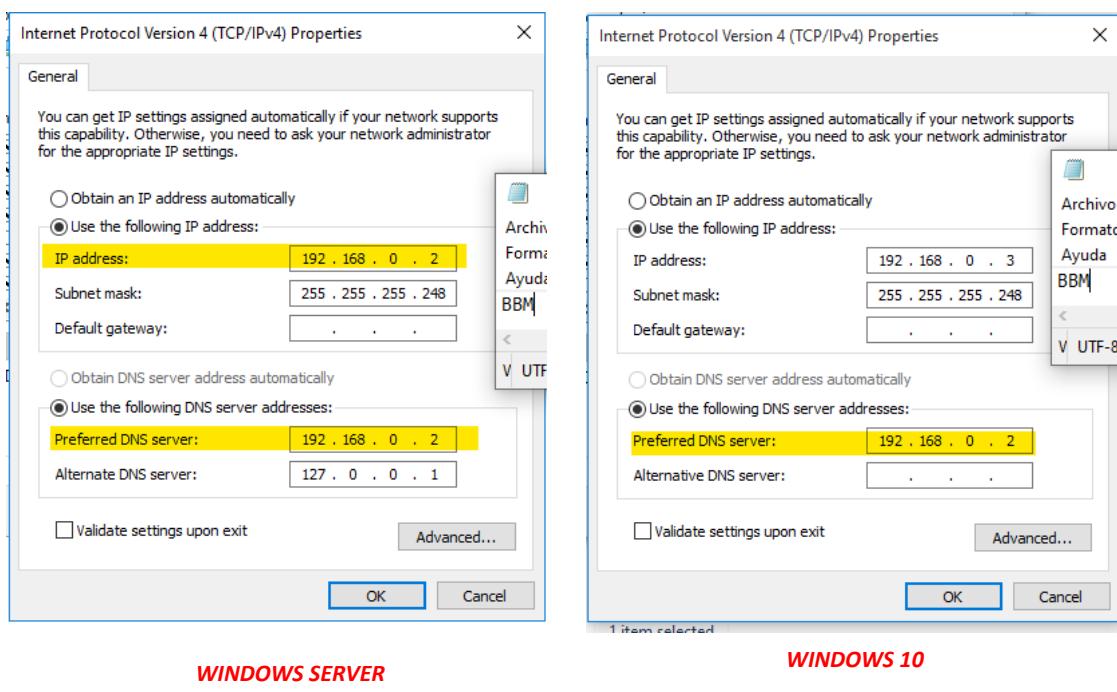
Básicamente, el proceso consistirá en realizar los siguientes pasos:

- Establecer las características de red, para que coincidan con las necesidades del dominio.
- Ajustar el nombre del equipo cliente.
- Unir el equipo al dominio.

Al final del proceso, se inicia sesión en el equipo cliente usando una cuenta de usuario de las que ya tenemos definidas en el dominio. Con esto se comprueba que el proceso se ha realizado correctamente.

Para conseguir que un cliente pueda unirse a un dominio, antes hay que asegurarse de que las características de su configuración de red sean coincidentes con las necesidades del dominio.

En las siguientes imágenes se ve la IP del servidor y cómo el cliente está en la red del servidor:



WINDOWS SERVER

WINDOWS 10

Ahora toca comprobar que la configuración de red es correcta. Hago un ping al servidor:

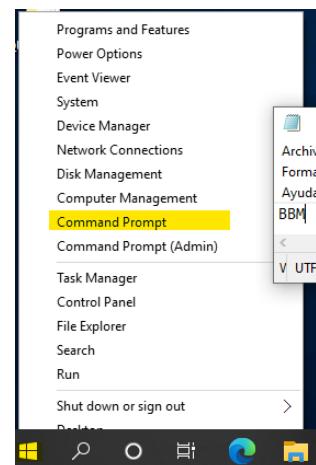
```
C:\Windows\system32>ping 192.168.0.2

Pinging 192.168.0.2 with 32 bytes of data:
Reply from 192.168.0.2: bytes=32 time=20ms TTL=128
Reply from 192.168.0.2: bytes=32 time=1ms TTL=128
Reply from 192.168.0.2: bytes=32 time=2ms TTL=128
Reply from 192.168.0.2: bytes=32 time=2ms TTL=128

Ping statistics for 192.168.0.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 1ms, Maximum = 20ms, Average = 6ms
```

Ahora abro una ventana de comandos y hacer un *ping* al Servidor. Utilizando el nombre del servidor en lugar de su dirección IP, se comprueba que la configuración *DNS* del cliente es correcta y que el servidor *DNS* del controlador de dominio está funcionando adecuadamente.

Como recordatorio, para acceder al cmd se hace clic de botón derecho sobre el símbolo de Windows y se selecciona *Command Prompt*.



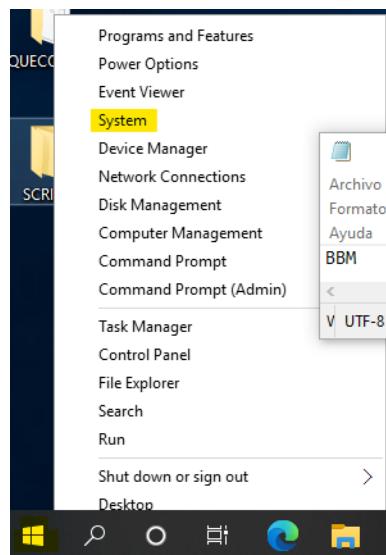
Una vez abierta la línea de comandos, se escribe `ping bbmaspace.local`. Si funciona se obtiene un resultado como el siguiente

```
C:\Windows\system32>ping bbmaspace.local

Pinging bbmaspace.local [192.168.0.2] with 32 bytes of data:
Reply from 192.168.0.2: bytes=32 time=7ms TTL=128
Reply from 192.168.0.2: bytes=32 time=1ms TTL=128
Reply from 192.168.0.2: bytes=32 time=1ms TTL=128
Reply from 192.168.0.2: bytes=32 time=152ms TTL=128

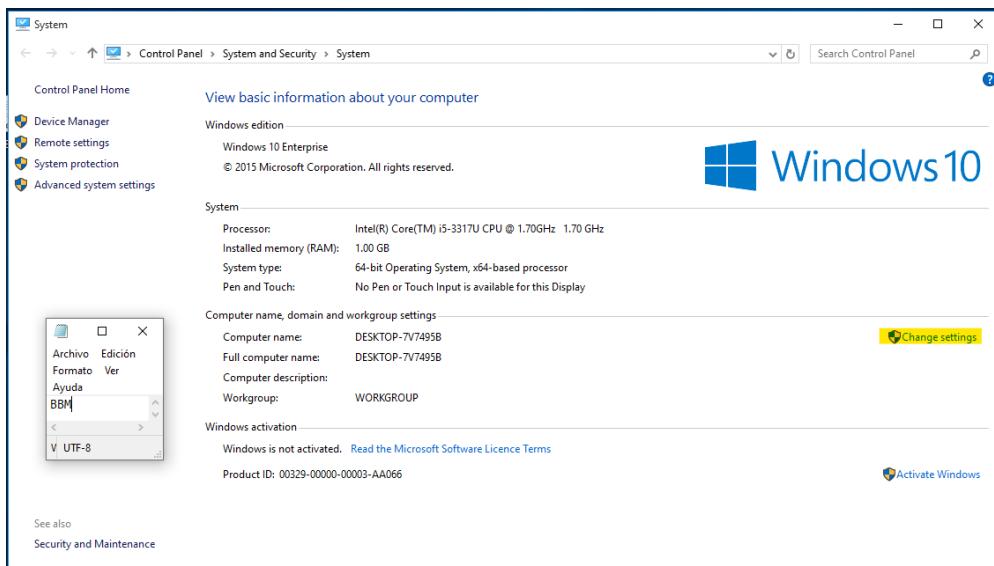
Ping statistics for 192.168.0.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 1ms, Maximum = 152ms, Average = 40ms
```

El siguiente paso consistirá en ajustar el nombre del equipo cliente, para que coincida con un nombre de equipo definido en el dominio. Al indicar el nombre del dominio, el sistema procederá a establecer el vínculo. Para comenzar, clic con el botón derecho del ratón, sobre el botón . Pinchamos en **System**.

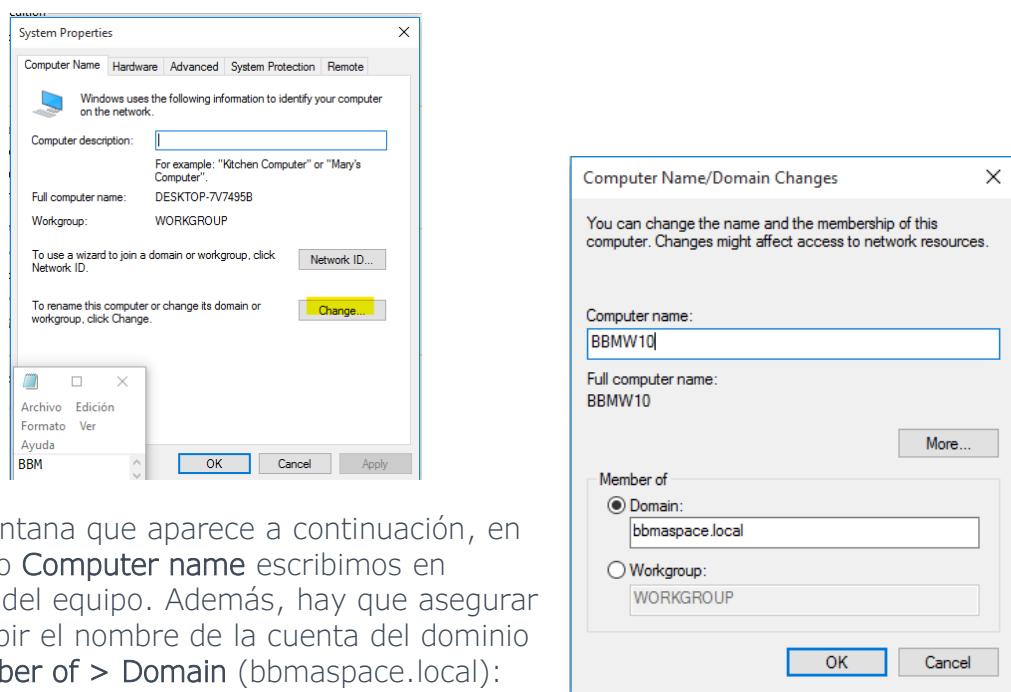


Al hacerlo, aparecerá una ventana con información del sistema. En la parte inferior, además de algunos detalles sobre el hardware del equipo, se encuentra su nombre (tanto el nombre *NetBIOS* como el nombre *DNS*), la descripción y el nombre del grupo de trabajo o dominio al que pertenece.

En la imagen se observa el nombre del equipo es el predeterminado que asigna *Windows* durante su proceso de instalación. Además, también pertenece al grupo de trabajo predeterminado (**WORKGROUP**), que se asigna a todos los equipos con *Windows* desde hace innumerables versiones. Para cambiar estos datos, clic en **Change Settings**:



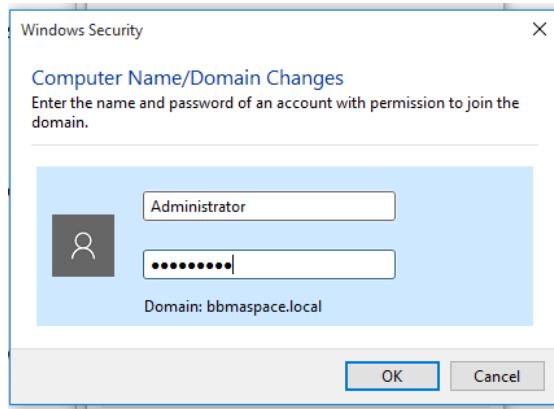
En la ventana **System Properties**, Está la opción **Change**; clicamos:



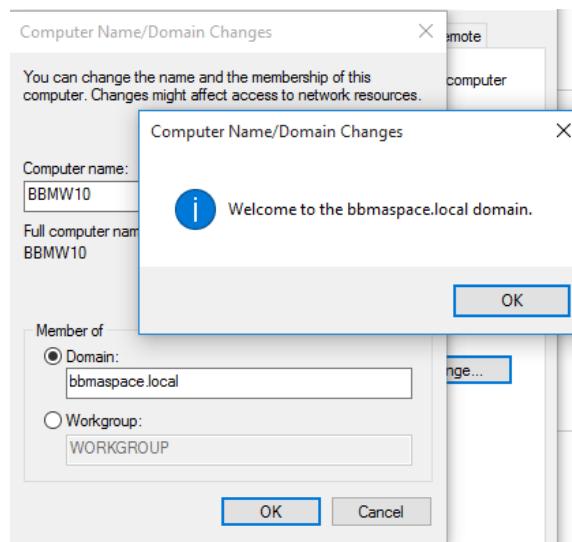
En la ventana que aparece a continuación, en el campo **Computer name** escribimos en nombre del equipo. Además, hay que asegurar de escribir el nombre de la cuenta del dominio en **Member of > Domain** (bbmaspace.local):

En ese momento, *Windows 10* busca en la red el dominio especificado. Si no lo encuentra, aparecerá un mensaje de aviso (tocaría revisar todo porque habría algo mal).

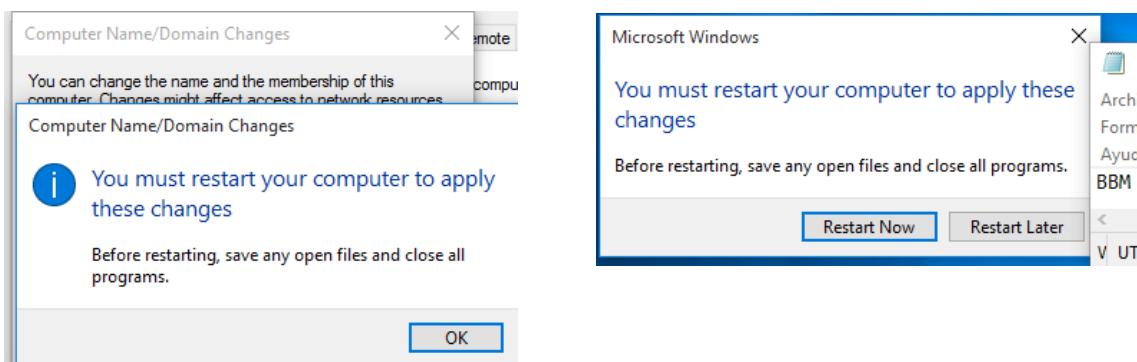
Si lo encuentra, sale un pop-up donde escribo **Administrator** y la **password** del dominio (**Abcd1234.**) En la parte de abajo del pantallazo se ve el Dominio: **bbmaspace.local**:



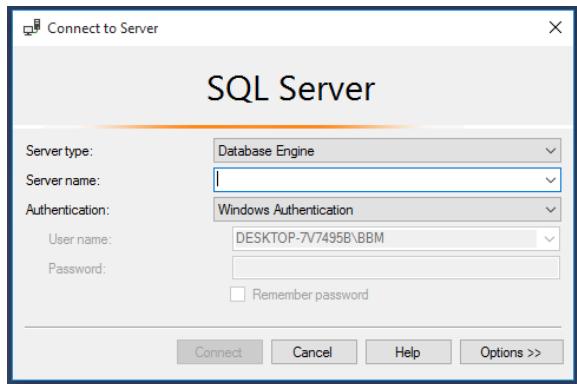
La ventana de autenticación se cierra y en su lugar aparece un mensaje indicando que el equipo se ha unido correctamente al dominio.



A continuación, aparece una nueva ventana informativa indicando que hay que reiniciar el equipo para que se apliquen los cambios. Cierro todos los programas, le doy a OK y a Restart Now.



Como va a ser la última vez que me conecte de manera local, dejo registrada una conexión con el cambio de nombre de equipo. Entro como **sa** con password **sa**.



Compruebo los logins y veo todos los que hay:

Login
##MS_PolicyEventProcessin
##MS_PolicyTsqlExecutionL
DESKTOP-7V7495B\BBM
NT AUTHORITY\SYSTEM
NT Service\MSSQLSERVER
NT SERVICE\SQLSERVERAGE
NT SERVICE\SQLTELEMETRY
NT SERVICE\SQLWriter
NT SERVICE\Winmgmt
sa

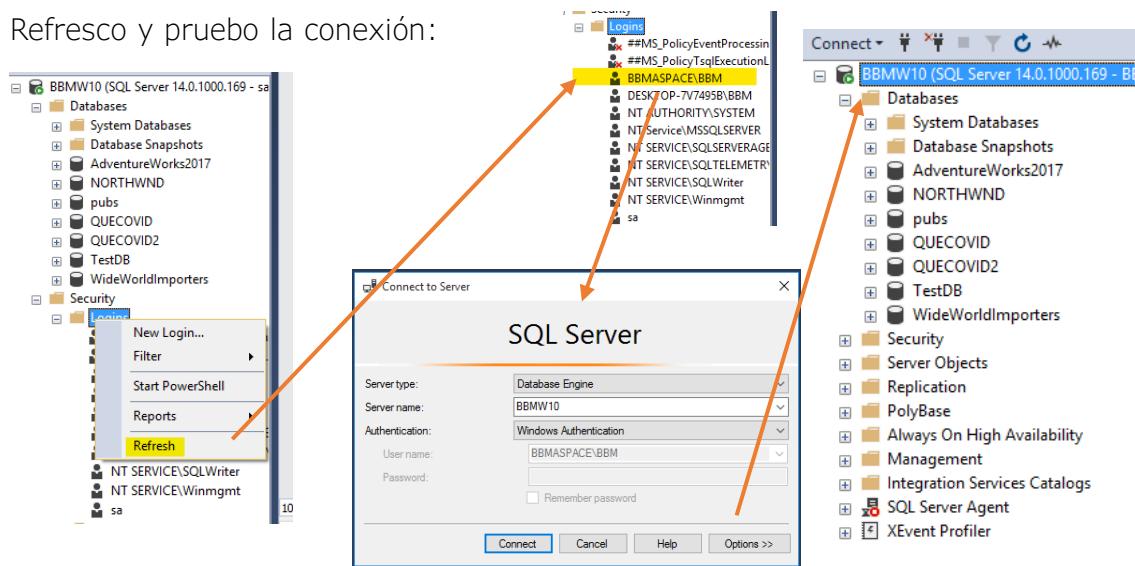
Ejecuto una query para crear un login con el usuario **BBMASPACE**.

```
USE [master]
GO

/***** Object: Login [DESKTOP-7V7495B\BBM]      Script Date:
21/12/2020 11:22:41 *****/
CREATE LOGIN [BBMASPACE\BBM] FROM WINDOWS WITH
DEFAULT_DATABASE=[master], DEFAULT_LANGUAGE=[us_english]
GO

ALTER SERVER ROLE [sysadmin] ADD MEMBER [BBMASPACE\BBM]
GO
```

Refresco y pruebo la conexión:

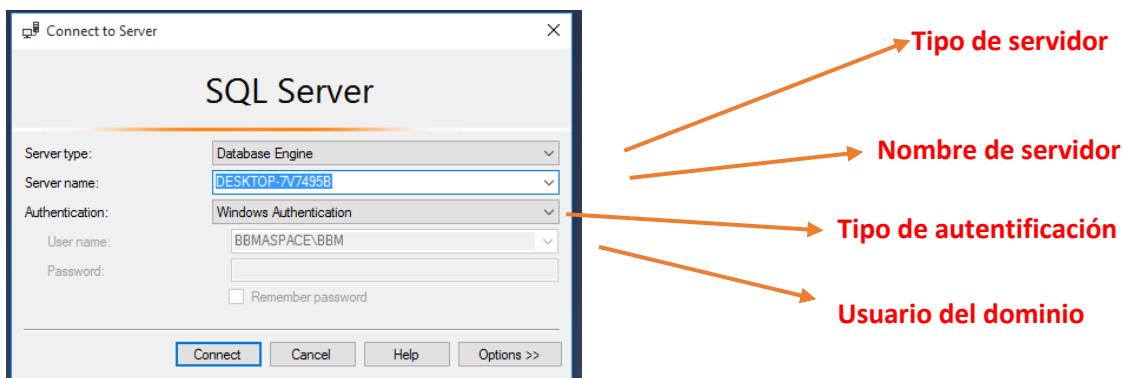


Acceder desde SQL Server a otras instancias de otros equipos del dominio

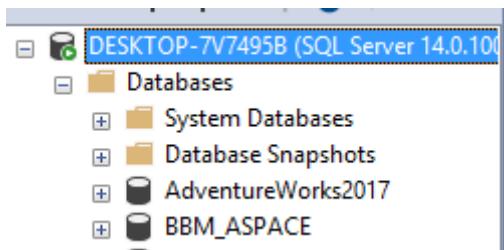
Después de unir todas las máquinas en el dominio, procedemos a conectarlas entre sí. Para esta maniobra tengo las tres encendidas a la vez.

Tanto en el Windows Server como en el cliente Windows 10 usamos el SSMS, mientras que en Ubuntu me paso al Azure Data Studio.

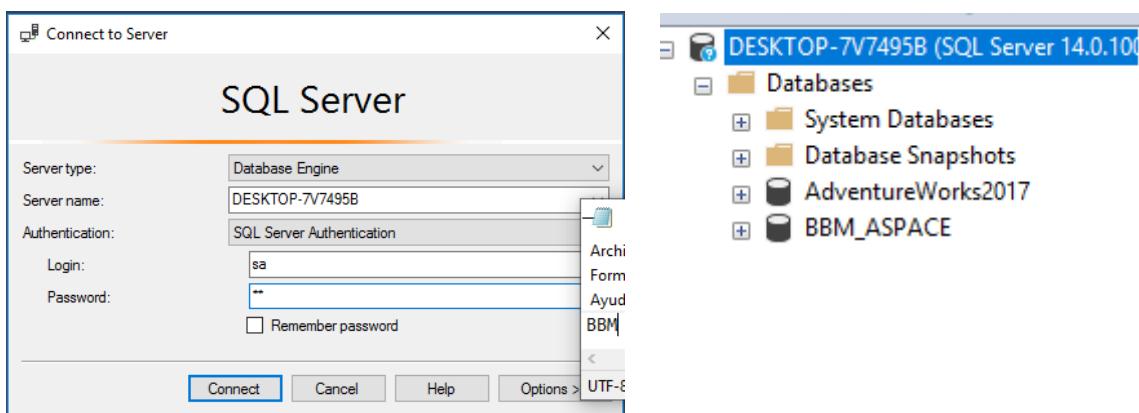
El sistema es el mismo de siempre → Ejemplo en Windows 10



Como se puede ver, conecta con la base:

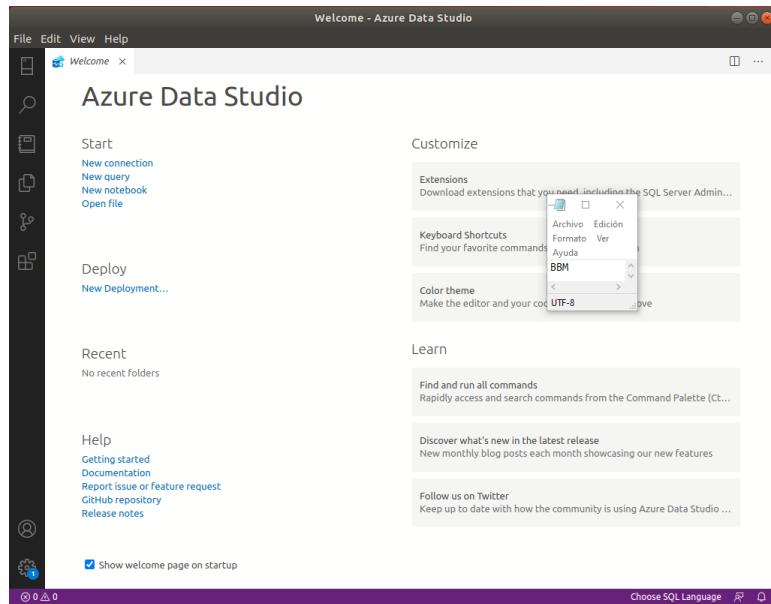


Ahora pruebo en Windows Server 16. La conexión que uso es con **sa** con **password sa** y la autenticación de **SQL Server**.



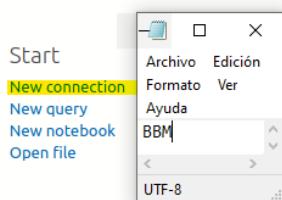
Por último, en *Ubuntu* entro en el **Azure Data Studio**. Aunque la explicación de todo viene más adelante... Lo suyo es meter todas las conexiones en el mismo apartado. Hago doble clic en el icono .

La página de inicio se ve de la siguiente manera:

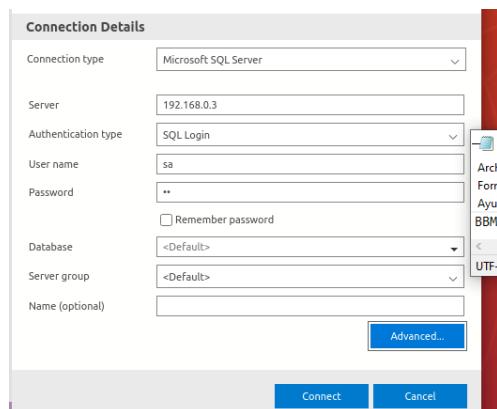


En la parte izquierda selecciono **New connection**:

Azure Data Studio



Se abre un desplegable donde pongo la conexión. Es exactamente igual que las otras, sólo que aquí lo hago con la **IP del cliente**:



Dejo constancia de las bases de datos para ver que está conectado al cliente:

The screenshot shows the Azure Data Studio interface with the title bar "192.168.0.3 - Azure Data Studio". The left sidebar has icons for Home, Databases (selected), Tables, Views, Procedures, Functions, and Scripts. The main pane displays a table of databases with columns: Name, Status, Size (MB), and Last backup. The table includes the following data:

Name	Status	Size (MB)	Last backup
master	ONLINE	7	
model	ONLINE	16	
msdb	ONLINE	21	
tempdb	ONLINE	24	
AdventureWorks2017	ONLINE	336	2017-10-27 02:35:38
BBM_ASPACE	ONLINE	30	
CAR	ONLINE	16	
Colegio_Medico	ONLINE	39	
NORTHWND	ONLINE	8	
PruebaFS	ONLINE	16	
pubs	ONLINE	16	
QUECOVID	ONLINE	16	
QUECOVID2	ONLINE	16	2020-11-14 08:40:55
SQLFileTable	ONLINE	16	
TestDB	ONLINE	16	
TRIALBBM	ONLINE	39	
WideWorldImporters	ONLINE	780	

At the bottom, it says "Choose SQL Language 192.168.0.3 : master" and shows connection statistics: 0 rows, 0 inserts, 0 updates, and 0 deletes.

Explicar Instalación Desatendida

Hay una situación que todo el mundo encuentra tarde o temprano: Se necesitan desplegar unas cuantas máquinas virtuales, y nadie se puede permitir el lujo de esperar durante 1 hora a que terminen de instalarse y configurarse. Lo que prima es la velocidad, sobre todo, entonces... ¿Cuál es la solución?

Al rescate vienen dos herramientas poderosas: **SysPrep** y los **Discos de Diferenciación**. Antes de entrar en materia, voy a resumir qué es y qué hace cada una de ellas:

¿Qué es SysPrep (System Preparation)? Es una herramienta que prepara una instalación de Windows (tanto de escritorio como Windows Server) para crear una imagen, permitiendo capturar una instalación personalizada. Esto lo consigue eliminando información específica del PC donde se ha instalado, "Generalizando" la instalación para que así se pueda instalar en otro PC. Con **SysPrep** se puede configurar el PC para que arranque la "*Out-of-Box Experience*" para que se encienda como si Windows estuviese recién instalado. Incluso se puede especificar un archivo **xml** con los datos necesarios para realizar una instalación desatendida (lo cual voy a hacer en este tema). Básicamente, es la herramienta que los fabricantes de portátiles usan para llenar con su software una instalación de Windows, y luego la generalizan para que podamos personalizarla como si estuviese recién instalada.

¿Qué es un Disco de diferenciación? Es un disco virtual que se utiliza para aislar cambios en los datos del disco o del sistema operativo instalado en él, manteniéndolos en un archivo separado. En este caso, un disco diferencial consta de dos partes: El disco "base", que se utiliza como datos origen que no van a cambiar, y el disco de diferenciación, que es básicamente un disco de tamaño dinámico en el que se escriben todos los nuevos datos. Una cualidad de especial interés es que se pueden crear múltiples discos de diferenciación a partir del mismo disco base.

A mayores de esto, usaremos la experiencia lista para usar (**OOBE**) que debe completar para configurar ajustes como la distribución del teclado, la cuenta y la configuración de privacidad.

Aunque el proceso es bastante sencillo, aún necesita pasar tiempo gestionando la instalación y respondiendo preguntas, lo que puede llevar mucho tiempo, especialmente si necesita realizar múltiples instalaciones en el lugar de trabajo. Para automatizar el proceso de instalación (mientras ahorra tiempo), se puede crear un archivo de respuesta con instrucciones para completar cualquier mensaje en pantalla que pueda integrar en un dispositivo de arranque. Luego, el programa de instalación de Windows puede leer para instalar automáticamente Windows 10. El nombre de este archivo es el **autounattend.xml**

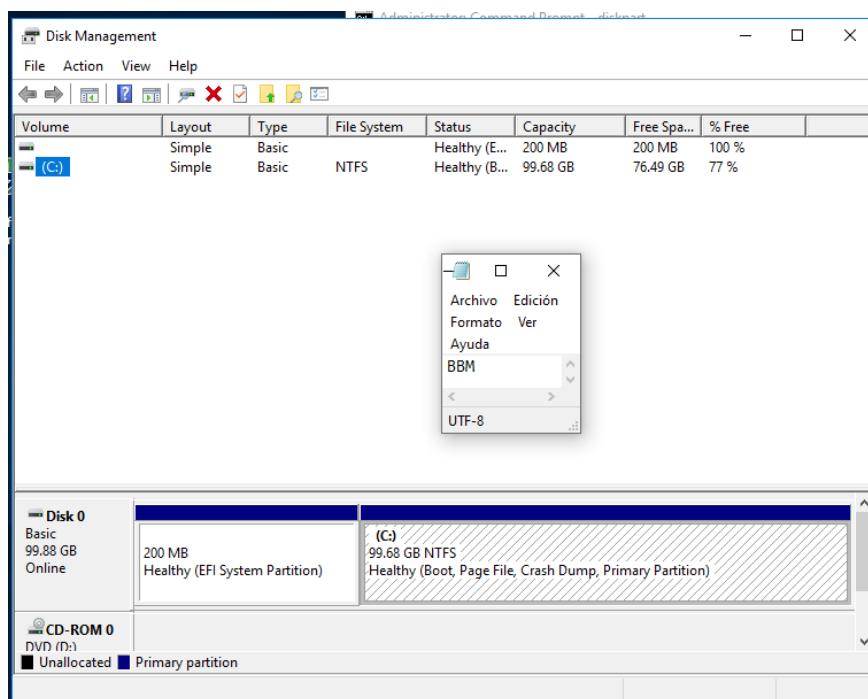
Entre las cosas que estoy nombrando para hacer la instalación, también será necesario el **Kit de evaluación y desarrollo de Windows (ADK)**.

Ahora que hay una introducción de lo que hace falta para hacer este trabajo, distinguiremos dos máquinas: **Reference/Source** y **Tech/Destiny**.

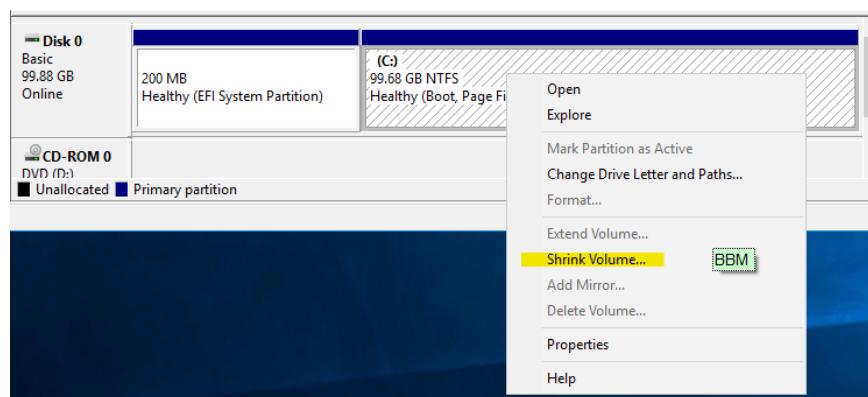
En la máquina source, haremos clic con el botón derecho sobre el símbolo de Windows  y pulsamos en el botón **Disk Management**.

Una vez dentro se observan dos partes del disco físico. Una que ocupa 200 MB, llamada **EFI**. EFI es una pequeña partición formateada FAT 32 en un dispositivo de almacenamiento de datos que se utiliza en computadoras que se adhieren a la interfaz de firmware extensible unificada. Se genera automáticamente cuando instala su sistema operativo, ya sea Windows o Mac OS. Cuando inicia su PC, el firmware UEFI carga archivos esenciales almacenados en el ESP para iniciar el sistema operativo instalado y varias utilidades necesarias para el inicio de su máquina. La partición del sistema EFI contiene cargadores de arranque, controladores de dispositivos, utilidades del sistema y algunos archivos de datos clave que son absolutamente necesarios para arrancar Windows.

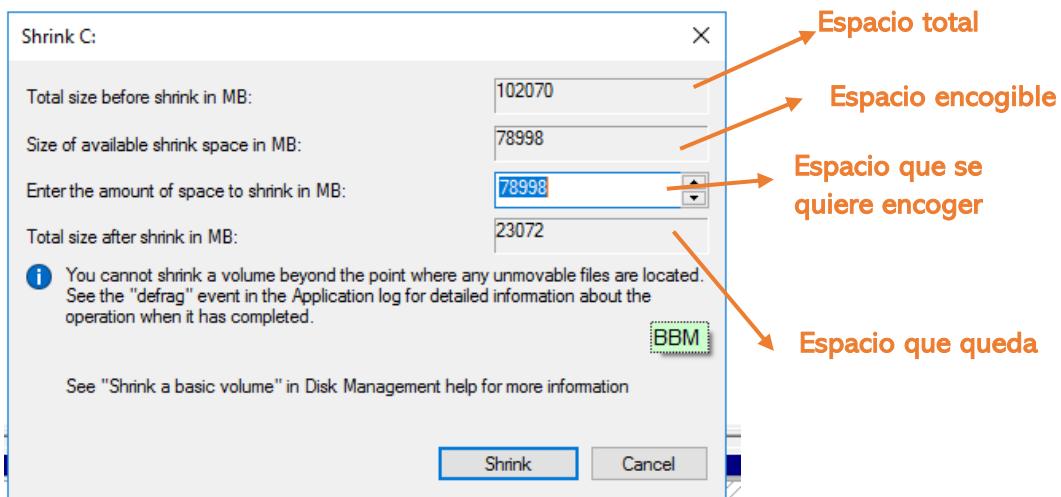
La otra sería nuestro disco de tipo **NTFS**, en el cuál guardamos el resto de información.



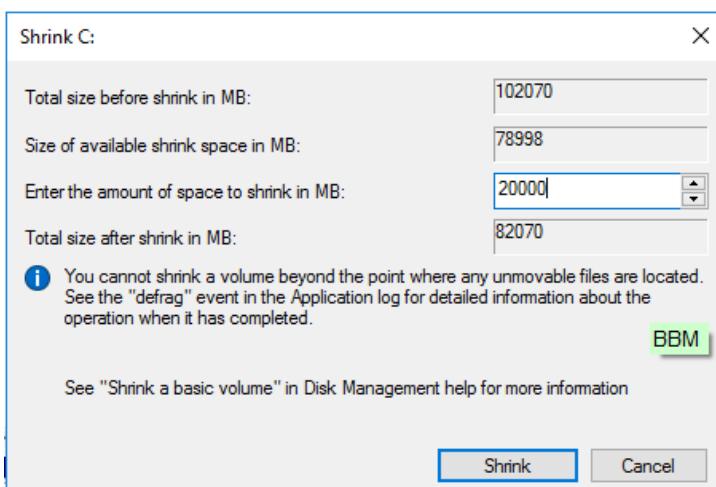
Con este disco lo que se va a hacer es **encoger** una parte (**shrink**), y dejarlo dividido en tres: 200 MB EFI, 80'15 GB, y otros 19'53 GB NTFS para **nuestra instalación desatendida**.



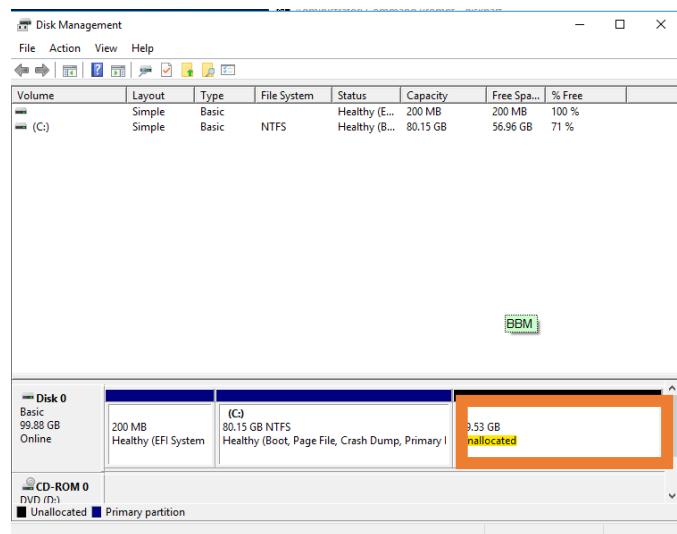
Se abre una ventana donde podemos darle la opción de encoger el disco en Enter the amount of space to shrink in MB:



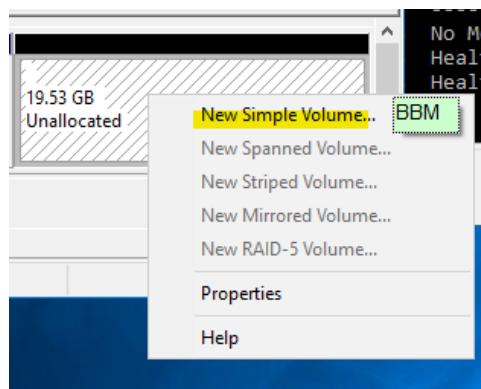
Lo que haré será poner unos 20000 (MB) que quiero que se encojan:



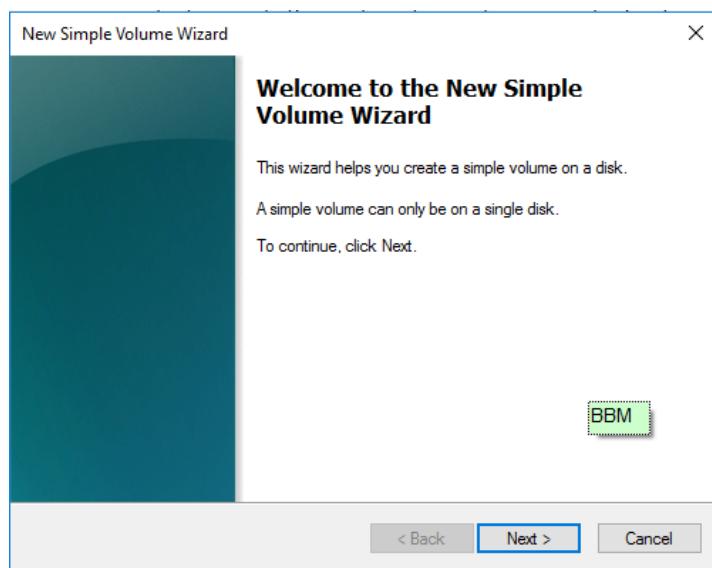
Donde antes había 99.98 GB, ahora hay dos particiones, una con 80.15 GB NTFS, y otro con 19.53 GB Unallocated.



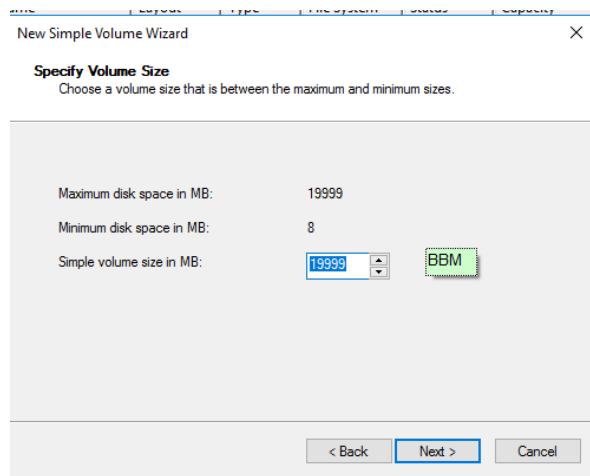
Es el turno de darle formato. Hago clic con el botón derecho sobre el “nuevo” disco y le doy a New Simple Volume:



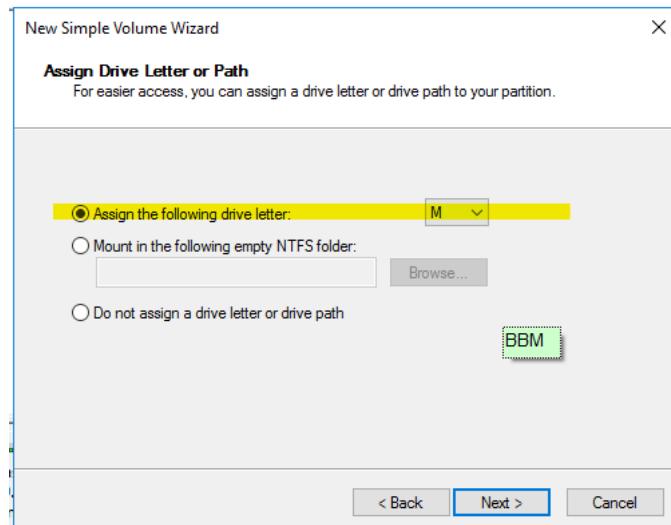
Se abre un asistente (*wizard*) cuya primera pantalla es de presentación.



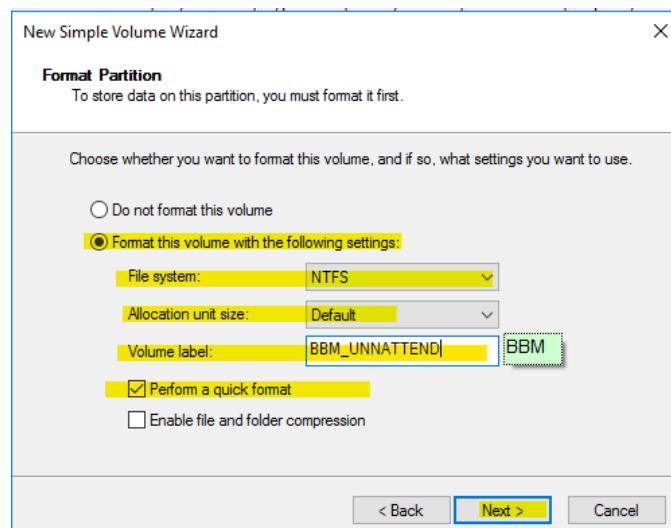
En la siguiente ventana especificamos el tamaño que va a tener este volumen (de los 20.00 GB se podría subdividir más aún). En este caso, uso los 20 GB.



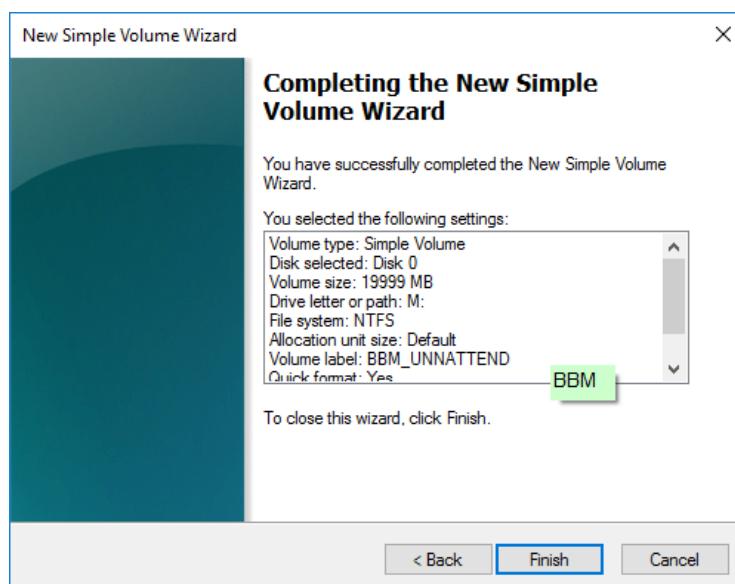
En la siguiente ventana, le asigno en la opción **Assign the following drive letter** una letra. Esto es realmente un poco aleatorio... Elegí la M porque es la inicial de mi nombre y así concuerda con lo que se pide en el proyecto.



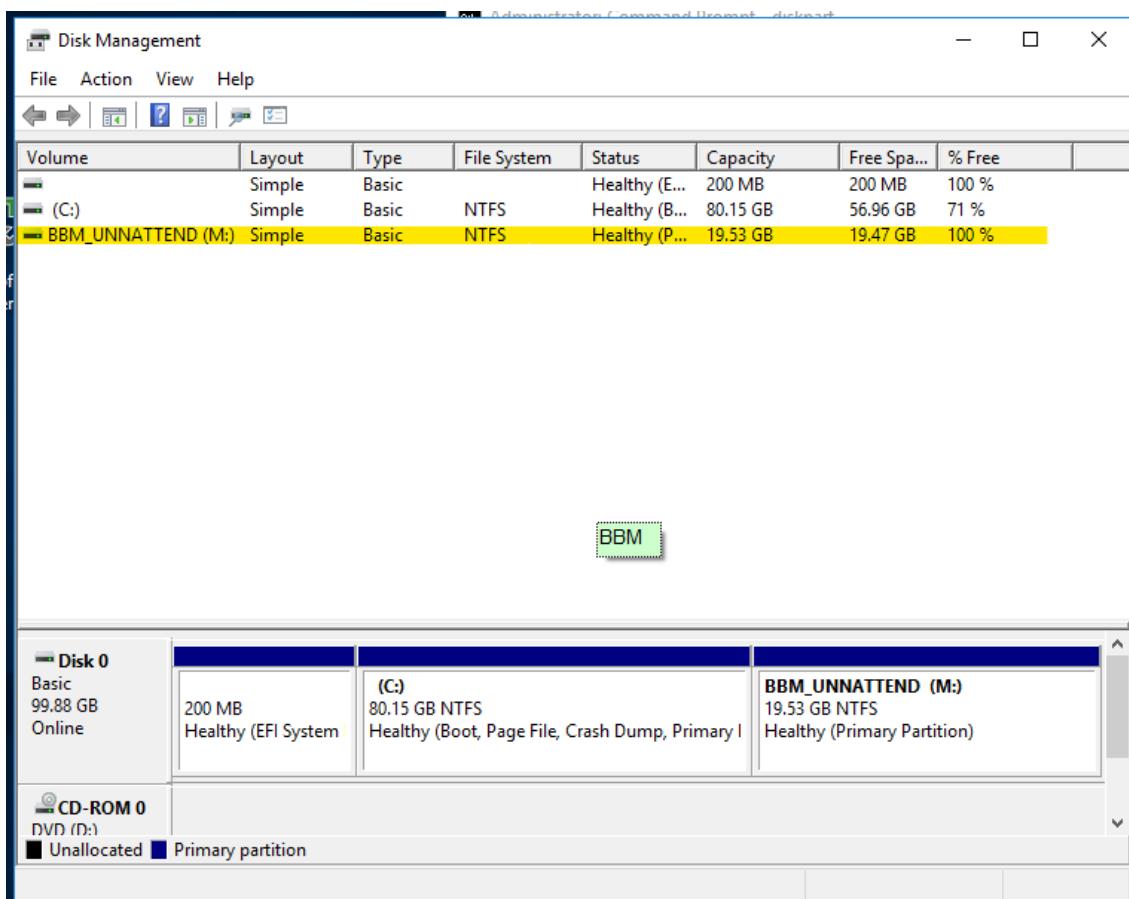
En la siguiente, le damos el formato al disco y un nombre: NTFS y BBM_UNNATTEND elegí:



La última pantalla como todo asistente, es un resumen. Le doy a Finish.



Como se ve en esta pantalla, queda dividido en dos, con un disco completamente utilizable con formato NTFS.



Ahora toca utilizar **Sysprep**. Es una herramienta de preparación del sistema que permite editar imágenes de Windows para llevarlas a un estado general. Este software está presente en todas las instalaciones de Windows, y es muy importante preparar un sistema de clonación y distribución en múltiples máquinas en la misma red. Por lo tanto, es una herramienta esencial para los administradores de TI, especialmente aquellos que administran redes con Active Directory.

De hecho, los sistemas operativos de Microsoft tienen identificadores de seguridad (SID) generados de manera única que se asignan a cada ordenador como parte de la configuración inicial de Windows. Cada vez que se instala Windows, se asigna un SID a esa instalación en particular. **Si hay más de un ordenador con el mismo SID en la misma red, esto puede causar problemas.**

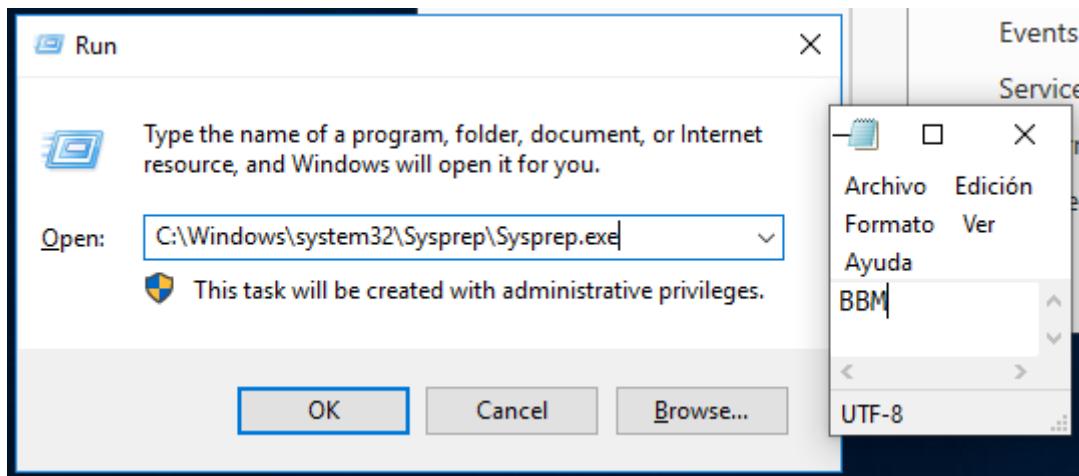
Para simplificar el despliegue de decenas o incluso cientos de estaciones de trabajo y servidores, muchos administradores de TI usan técnicas de clonación de un sistema operativo de muestra que está configurado con las aplicaciones, software y configuraciones correctas para distribuirlo a otros ordenadores utilizando una imagen de disco.

Aquí es donde el problema de la duplicación de SID se vuelve crítico. Clonar de una ordenador físico o virtual sin regenerar el SID como parte del proceso de clonación no es compatible con Microsoft.

Para resolver este problema, Microsoft creó la herramienta “Preparación del sistema” o “Sysprep”. Esta herramienta se puede utilizar para eliminar información específica del sistema operativo, de modo que se pueda utilizar para volver a crear imágenes en varias computadoras.

Claramente, esta herramienta se debe usar única y exclusivamente en un sistema operativo recién instalado, con el único propósito de clonarlo y luego instalarlo en otras máquinas a través de una imagen.

Usar Sysprep es muy simple. Se encuentra en la carpeta C:\Windows\System32\Sysprep\Sysprep.exe, y podemos ejecutarlo desde el prompt usando esta simple línea de comando:

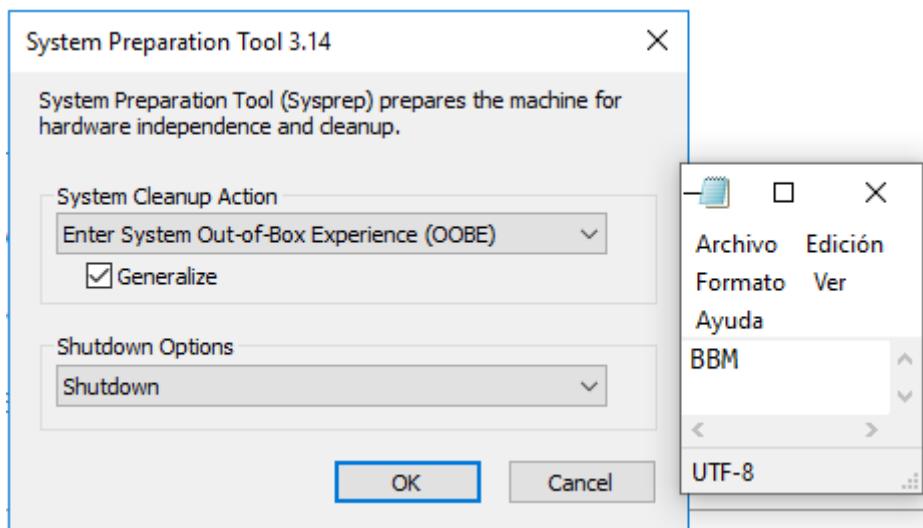


Se abre el **asistente Sysprep**, que consiste en dos partes: Enter System Out-of-Box Experience (OOBE) y las opciones de apagado (Shutdown Options). La herramienta GUI ofrece solo 3 opciones posibles:

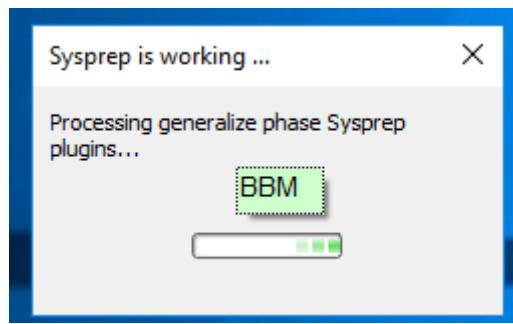
- System Cleanup Action
 - Enter System Out-of-box Experience (OOBE) → Esta es la función más utilizada de Sysprep y, básicamente, configura el servidor como si se hubiera iniciado por primera vez.
 - Enter System Audit Mode
 - El modo Audit iniciará automáticamente el ordenador como administrador local
 - La cuenta Administrador se desactiva automáticamente después del primer inicio de sesión

- El Servidor siempre se iniciará en modo Audit hasta que esté configurado para iniciarse en OOBE
 - Si se inicia un protector de pantalla protegido con contraseña, quedarás “bloqueado” fuera del servidor sin ninguna posibilidad alternativa
 - Este modo se usa para crear versiones altamente personalizadas del sistema operativo generalmente junto con herramientas de terceros.
- Generalize
 - Generaliza el SID – fundamental para la implementación de imágenes
 - Shutdown Options
 - Quit – cierra *Sysprep*
 - Reboot – una vez completado el *Sysprep*, reinicia el servidor
 - Shutdown – una vez completado el *Sysprep*, apaga el servidor

Ejecuto *Sysprep* con las opciones OOBE, Generalize y Shutdown habilitadas. Seleccionar tal como se indica a continuación:



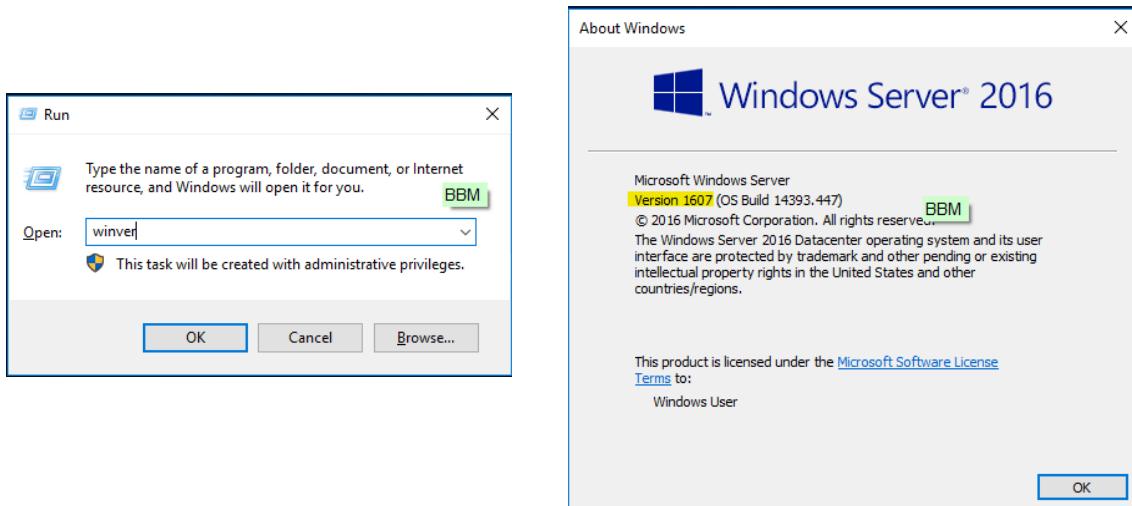
Una vez clico en OK, carga y se apaga.



El siguiente paso es instalar Windows ADK, el cual contiene un conjunto de herramientas para personalizar las imágenes de Windows para una implementación a gran escala. No solo puede utilizar las herramientas para probar la calidad y el rendimiento de su sistema y las aplicaciones que se ejecutan en él.

Lo primero será instalarlo en la máquina **Tech**. Lo descargamos de la página web <https://docs.microsoft.com/en-us/windows-hardware/get-started/adk-install>. Una vez dentro, vemos que hay un montón de versiones. Para saber cuál es la que

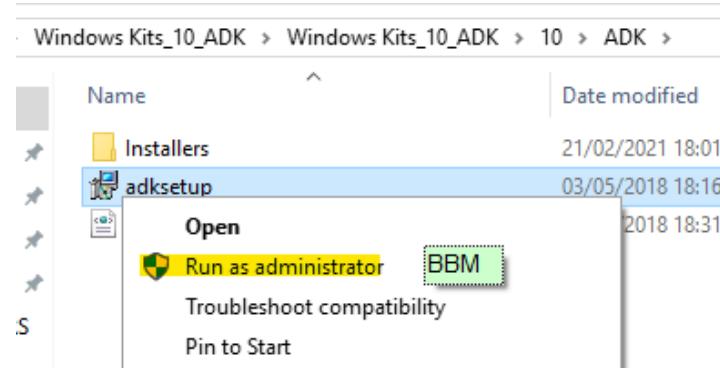
tengo que meter, tocamos las teclas **Windows + R** y escribimos **winver**. En la ventana que se abre vemos la versión de Windows que tenemos y que nos indicará la **versión de ADK** que debemos descargar.



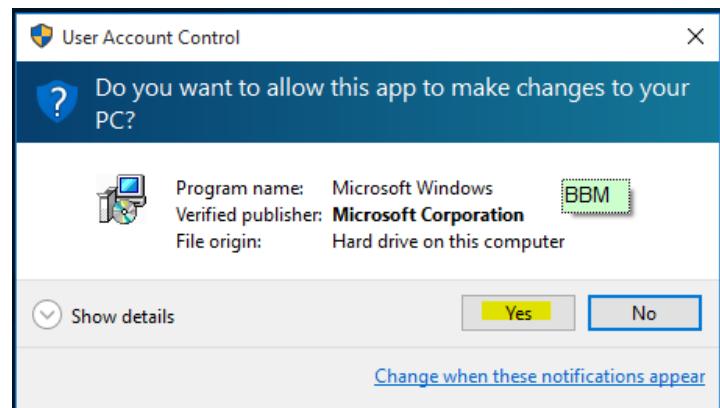
Si nos fijamos bien, las versiones que hay son las de Windows de 10. Aunque la máquina que estoy explicando es de **Windows Server 16**, nos vale, pues la base de Sistema Operativo que tiene es Windows 10, así que descargo esta versión.

[ADK for Windows 10, version 1607](#) [What's new in ADK for Windows 10, version 1607.](#) [For IoT Core for Windows 10, version 1607, also download the IoT Core Add-Ins v2.0 \(14393_v1\)](#).

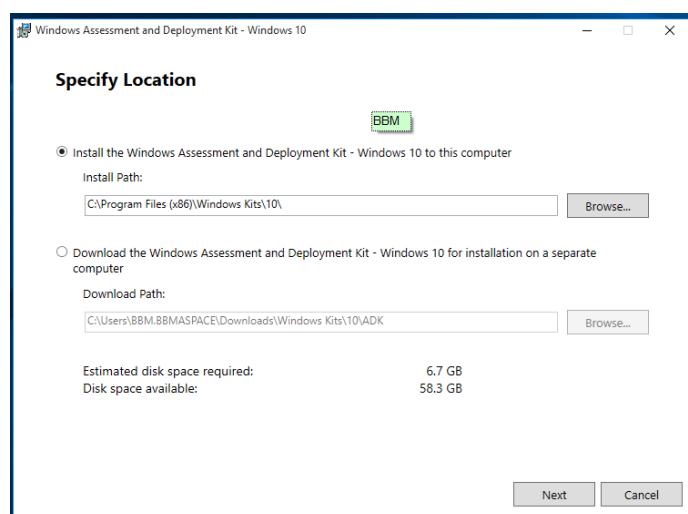
Una vez hecho, me voy al Setup ubicado en la ruta donde lo descargo y lo ejecuto como **Administrador**.



Acepto que comience la instalación.



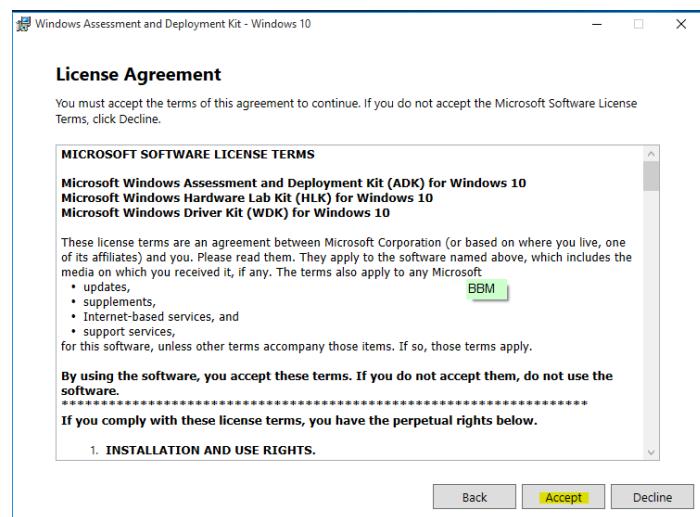
En la primera página se le instala la ruta por defecto. Le doy a Siguiente:



En el Windows Kits Privacy nos pregunta si queremos enviar datos a Microsoft. Le decimos que no:



Aceptamos la licencia:



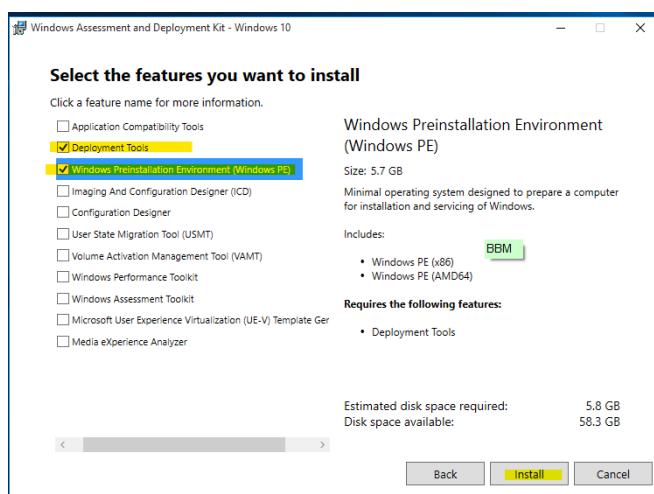
En las siguientes ventanas se incluyen las siguientes características. Al instalar Windows 10 ADK para SCCM, puede seleccionar las siguientes funciones para instalarlas.

- Application Compatibility Tools
- Deployment Tools
- Windows Preinstallation Environment (Windows PE)
- Imaging and Configuration Designer (ICD)
- Configuration Designer
- User State Migration Tool (USMT)
- Volume Activation Management Tool (VAMT)
- Windows Performance Toolkit
- Windows Assessment Toolkit
- Microsoft User Experience Virtualization (UE-V)
- Application Virtualization Sequencer (App-V)
- Microsoft Application Virtualization Auto Sequencer (App-V)
- Media experience Analyzer

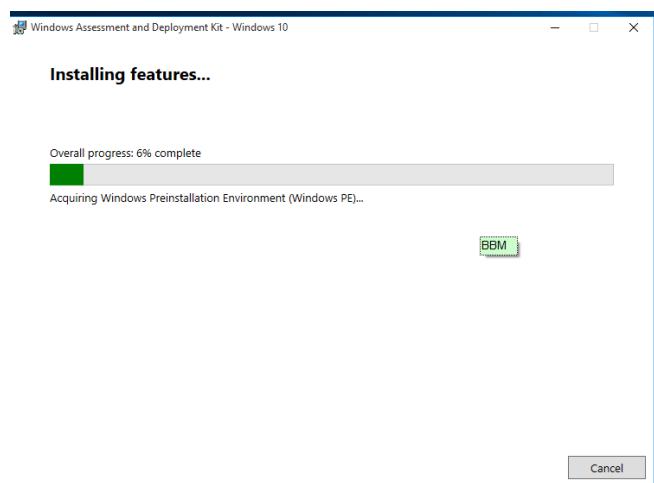
En la siguiente tenemos que elegir las aplicaciones, yo elegiré las dos siguientes:

- **Deployment Tools.** Elevan los servicios de actualización existentes, automatizan las tareas de implementación, implementan las mejores prácticas de seguridad y monitorean la actividad del usuario y el funcionamiento de las aplicaciones. Sin estas herramientas, los equipos de TI pasan horas implementando y rastreando implementaciones manualmente. Esto le quita tiempo y recursos a otras tareas importantes.
- **Entorno de preinstalación de Windows (Windows PE).** Es una versión "Lite" de Windows 10 y se creó para actuar como una plataforma de implementación. Windows PE reemplaza los discos de inicio de DOS o Linux que se encontraban las soluciones de implementación de la última década.

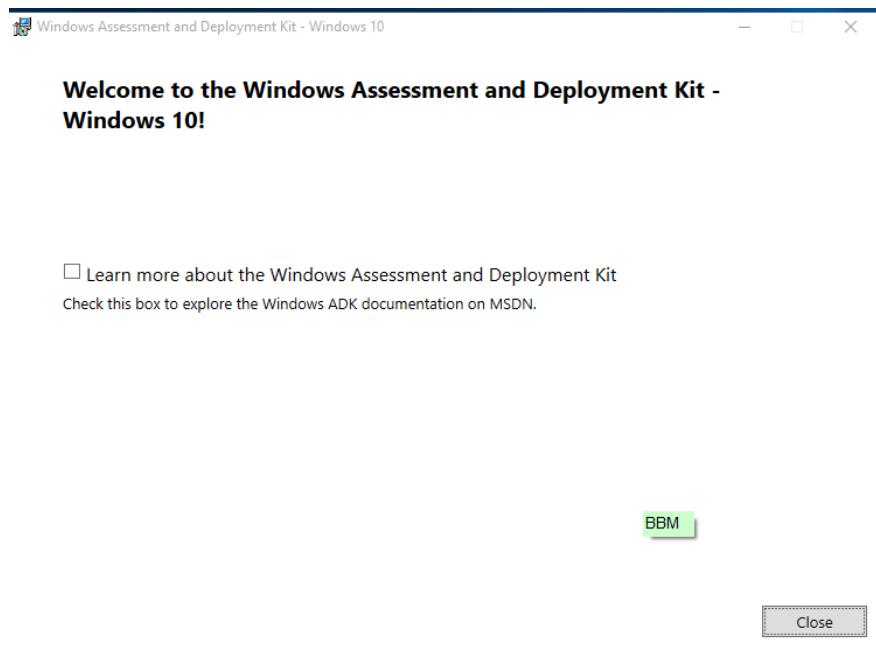
Lo importante que debe saber sobre Windows PE es que, al igual que el sistema operativo, necesitan drivers para al menos dispositivos de almacenamiento y red en cada equipo. Afortunadamente, Windows PE incluye los mismos drivers que el sistema operativo completo Windows 10, lo que significa que gran parte del hardware funcionará fuera de la caja.



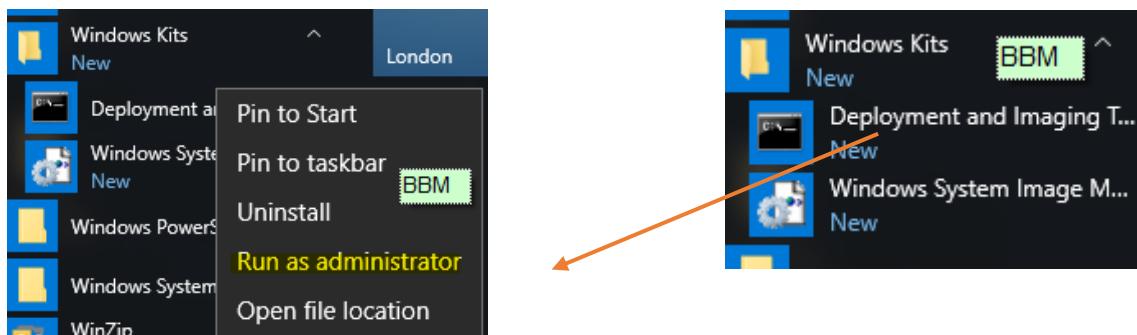
Una vez le das a **Install**, empiezan a cargar:



Al finalizar le damos a Close, y el resultado es el siguiente:



Ahora se empieza con la que es la verdadera instalación desatendida. Primero ejecutamos como administrador Deployment and Imaging Tools Environment.



En esta parte se va a preparar la creación de imagen de Windows PE. Para ello, ejecuto el comando `copype amd64 c:\Win10_Serv16PE`. ¿Qué es lo que hago con esto? Pues a partir de una imagen de Windows en el CD, se crea ese archivo WinPE:

```
cmd Select Administrator: Deployment and Imaging Tools Environment
C:\Program Files (x86)\Windows Kits\10\Assessment and Deployment Kit\Deployment Tools>copype amd64 c:\Win10_Serv16PE
=====
Creating Windows PE customization working directory
    BBM
=====
C:\Program Files (x86)\Windows Kits\10\Assessment and Deployment Kit\Windows Preinstallation Environment\amd64\Media\bootmgr
C:\Program Files (x86)\Windows Kits\10\Assessment and Deployment Kit\Windows Preinstallation Environment\amd64\Media\bootmgr.efi
C:\Program Files (x86)\Windows Kits\10\Assessment and Deployment Kit\Windows Preinstallation Environment\amd64\Media\bg-
bg\bootmgr.efi.mui
C:\Program Files (x86)\Windows Kits\10\Assessment and Deployment Kit\Windows Preinstallation Environment\amd64\Media\Boo-
t\BCD
C:\Program Files (x86)\Windows Kits\10\Assessment and Deployment Kit\Windows Preinstallation Environment\amd64\Media\Boo-
t\boot.sdi
C:\Program Files (x86)\Windows Kits\10\Assessment and Deployment Kit\Windows Preinstallation Environment\amd64\Media\Boo-
t\bootfix.bin
C:\Program Files (x86)\Windows Kits\10\Assessment and Deployment Kit\Windows Preinstallation Environment\amd64\Media\Boo-
t\memtest.exe
C:\Program Files (x86)\Windows Kits\10\Assessment and Deployment Kit\Windows Preinstallation Environment\amd64\Media\Boo-
t\bg-bg\bootmgr.exe.mui
C:\Program Files (x86)\Windows Kits\10\Assessment and Deployment Kit\Windows Preinstallation Environment\amd64\Media\Boo-
t\cs-cz\bootmgr.exe.mui
C:\Program Files (x86)\Windows Kits\10\Assessment and Deployment Kit\Windows Preinstallation Environment\amd64\Media\Boo-
t\cs-cz\memtest.exe.mui
C:\Program Files (x86)\Windows Kits\10\Assessment and Deployment Kit\Windows Preinstallation Environment\amd64\Media\Boo-
...
C:\Program Files (x86)\Windows Kits\10\Assessment and Deployment Kit\Windows Preinstallation Environment\amd64\Media\uk-
ual\bootmgr.efi.mui
C:\Program Files (x86)\Windows Kits\10\Assessment and Deployment Kit\Windows Preinstallation Environment\amd64\Media\zh-
cn\bootmgr.efi.mui
C:\Program Files (x86)\Windows Kits\10\Assessment and Deployment Kit\Windows Preinstallation Environment\amd64\Media\zh-
tw\bootmgr.efi.mui
155 File(s) copied
    1 file(s) copied.
    1 file(s) copied.
    1 file(s) copied.
    BBM
Success

C:\Program Files (x86)\Windows Kits\10\Assessment and Deployment Kit\Deployment Tools>
```

Una vez finalizado el paso anterior, introduzco el siguiente comando, embebiendo dentro de la ruta real a la carpeta donde creó los archivos PE, y la letra de unidad C: con la letra de unidad real de su unidad USB: **MakeWinPEMedia /ISO C:\Win10_Serv16PE C:\Win10_Serv16PE\Win10_Serv16PE.iso**

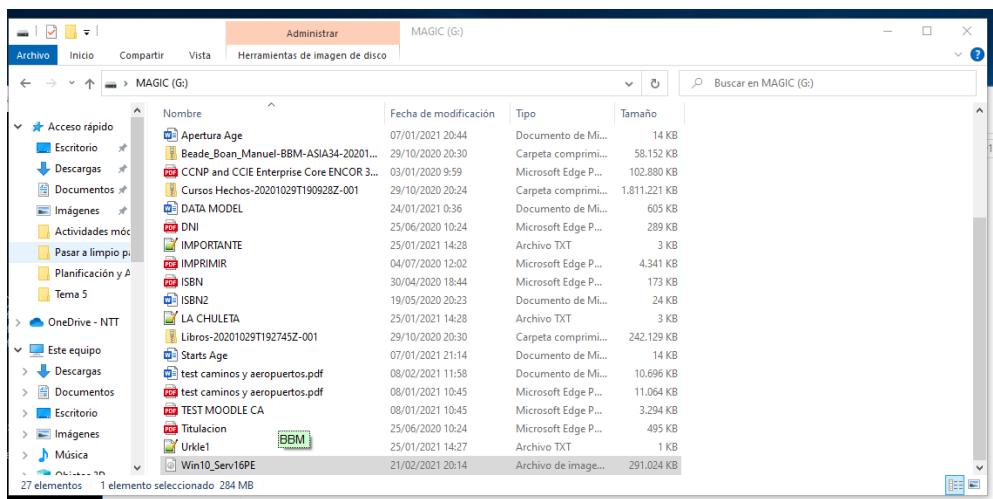
```
C:\Program Files (x86)\Windows Kits\10\Assessment and Deployment Kit\Deployment Tools>MakeWinPEMedia /ISO C:\Win10_Serv16PE C:\Win10_Serv16PE\Win10_Serv16PE.iso
Creating C:\Win10_Serv16PE\Win10_Serv16PE.iso...
    BBM
100% complete
Success

C:\Program Files (x86)\Windows Kits\10\Assessment and Deployment Kit\Deployment Tools>
```

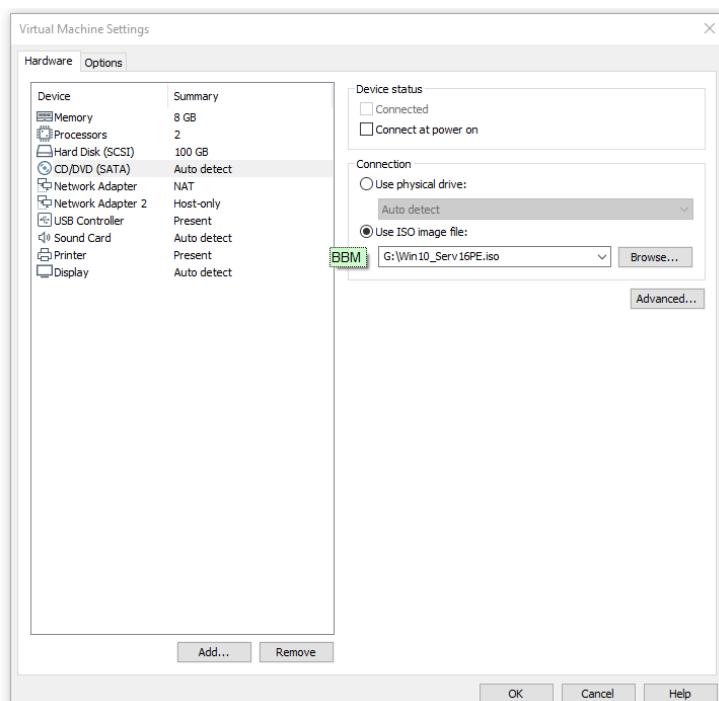
Como se ve, el resultado es algo así:

	Name	Date modified	Type	Size
ss	fwfiles	2/21/2021 8:05 PM	File folder	
js	media	2/21/2021 8:05 PM	File folder	
its	mount	2/21/2021 8:05 PM	File folder	
	Win10_Serv16PE	2/21/2021 8:14 PM	Disc Image File	291,024 KB

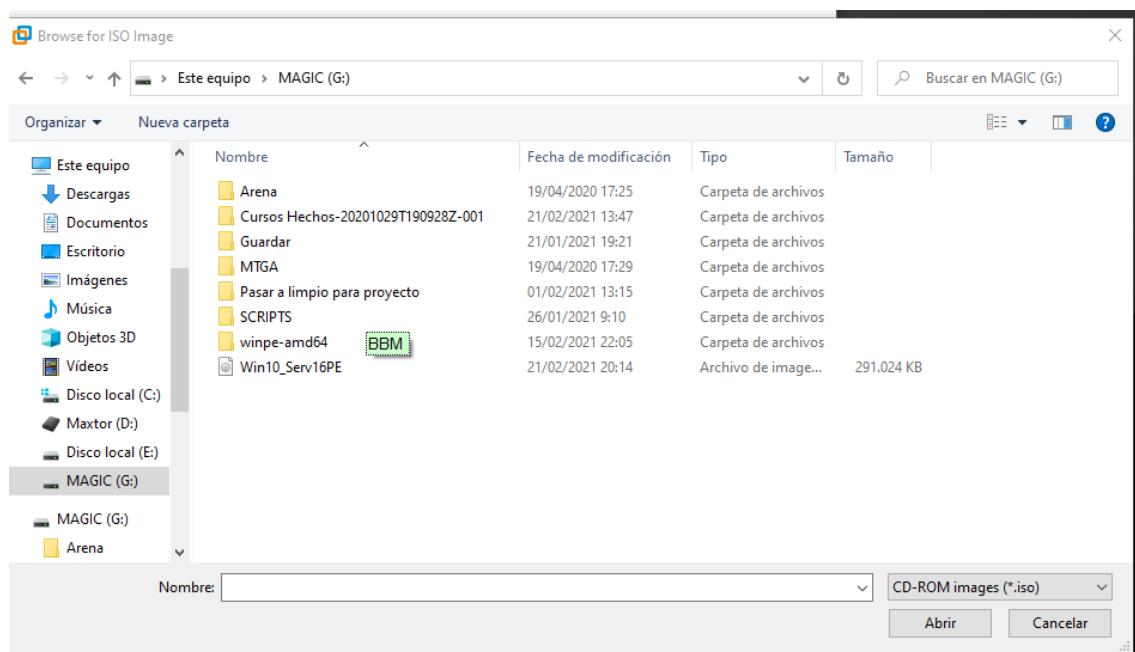
Ahora me copio esta imagen creada en un USB:



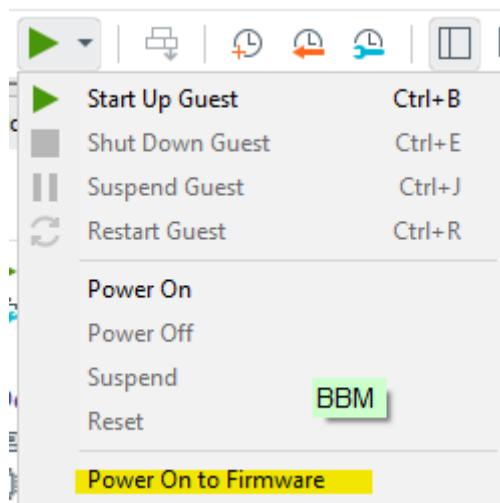
Ahora volvemos a Reference, en la cual editaremos las Virtual Machine Settings. Ahí, en la parte de CD/DVD (SATA), cambiaremos a la opción Use ISO image file:



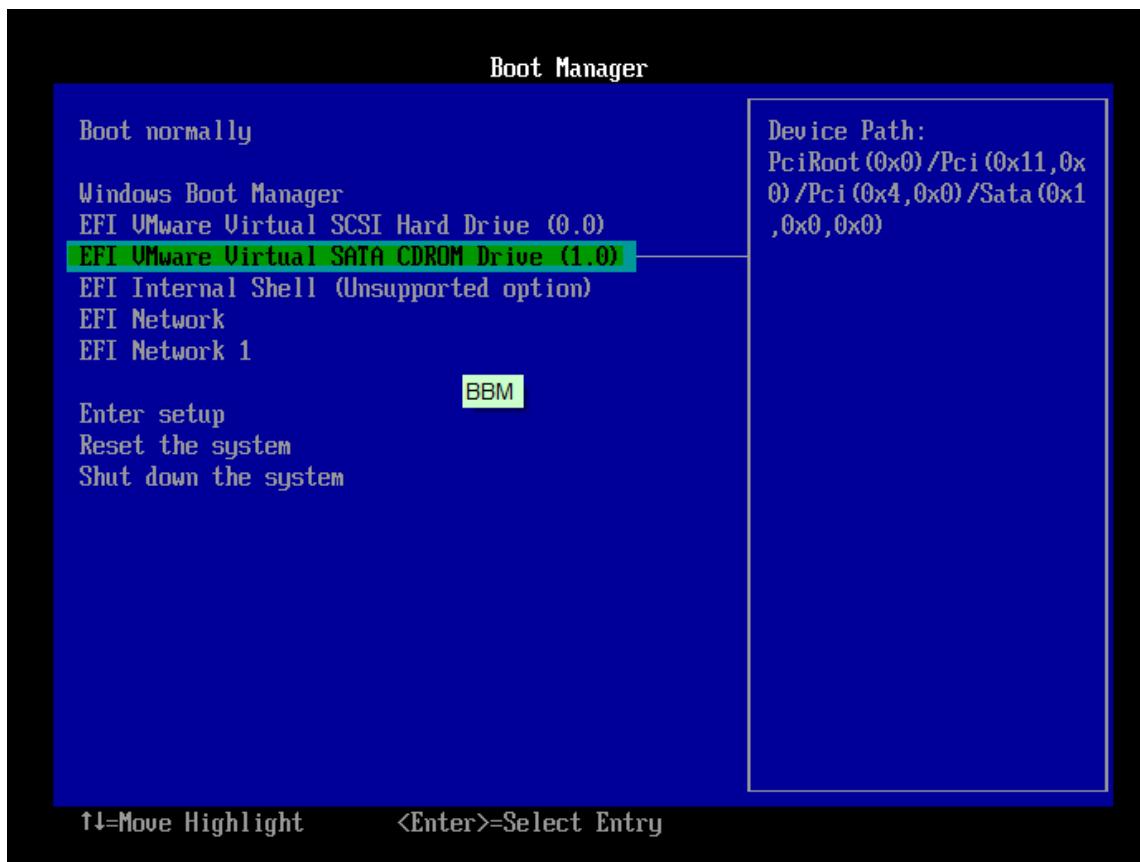
Encendemos la Reference metiendo antes la ISO creada en el CD/DVD.



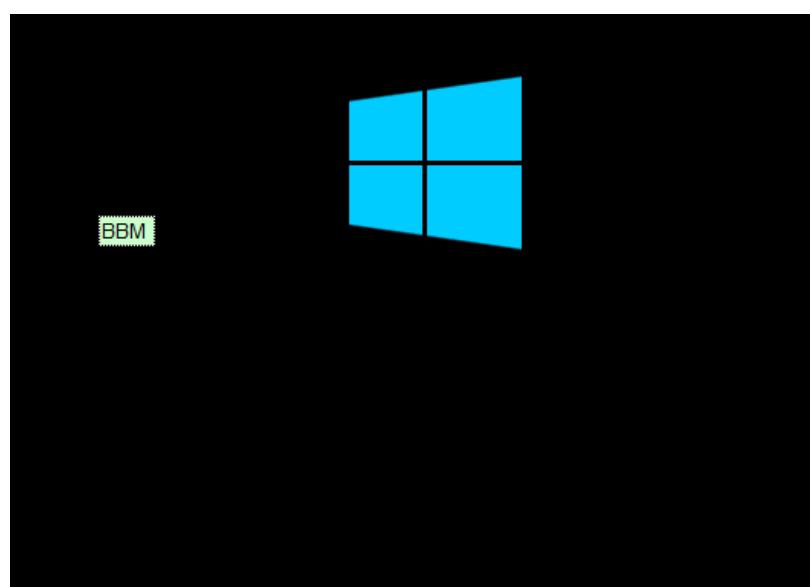
Una vez hecho esto, vamos al símbolo  y desplegamos en , eligiendo Power On to Firmware.



Ahora el inicio es en el **Boot Manager** en vez de directamente. Esto quiere decir que le voy a decir al sistema desde donde quiero que inicie. Le indico que lo haga desde el **EFI VMware Virtual SATA CDROM Drive (1.0)**.



Vemos que carga Windows



Vemos que se abre un terminal y se observa que la letra que pone X en vez de M, puede que hayan cambiado.

```
X:\windows\system32>wpeinit
X:\windows\system32>-
```

Comprobamos esto con el comando [diskpart](#), y vemos lo siguiente:

- E → nuestro dvd
- C → es nuestro disco local
- D → es la partición creada
- Hidden → La oculta es la FAT32 ES LA DE ARRANQUE

```
X:\windows\system32>diskpart
Microsoft DiskPart version 10.0.17134.1
Copyright (C) Microsoft Corporation.
On computer: MININT-HDN0KOI
DISKPART> list volume
  Volume ### Ltr Label Fs Type Size Status Info
  -----
  Volume 0 E DVD_ROM UDF DVD-ROM 284 MB Healthy
  Volume 1 C          NTFS Partition 80 GB Healthy
  Volume 2 D BBM_UNNATTE NTFS Partition 19 GB Healthy
  Volume 3          FAT32 Partition 200 MB Healthy   Hidden
DISKPART>
```

Ejecutamos para comprimir imagen, para ello usamos el comando [dism.exe /Capture-Image /Compress:max /Imagefile:D:\install.wim /Capturedir:C\ /Name:WS16x64_ASPACE](#). El comando Compress:max lo que hace es comprimir al máximo esa imagen

```
X:\windows\system32>dism.exe /Capture-Image /Compress:max /Imagefile:D:\install.wim /Capturedir:C\ /Name:WS16x64_ASPACE
Deployment Image Servicing and Management tool
Version: 10.0.17134.1
Saving image
[=====100.0%=====]
The operation completed successfully.
X:\windows\system32>
```

Vemos que en el directorio D:\ nos creó un archivo

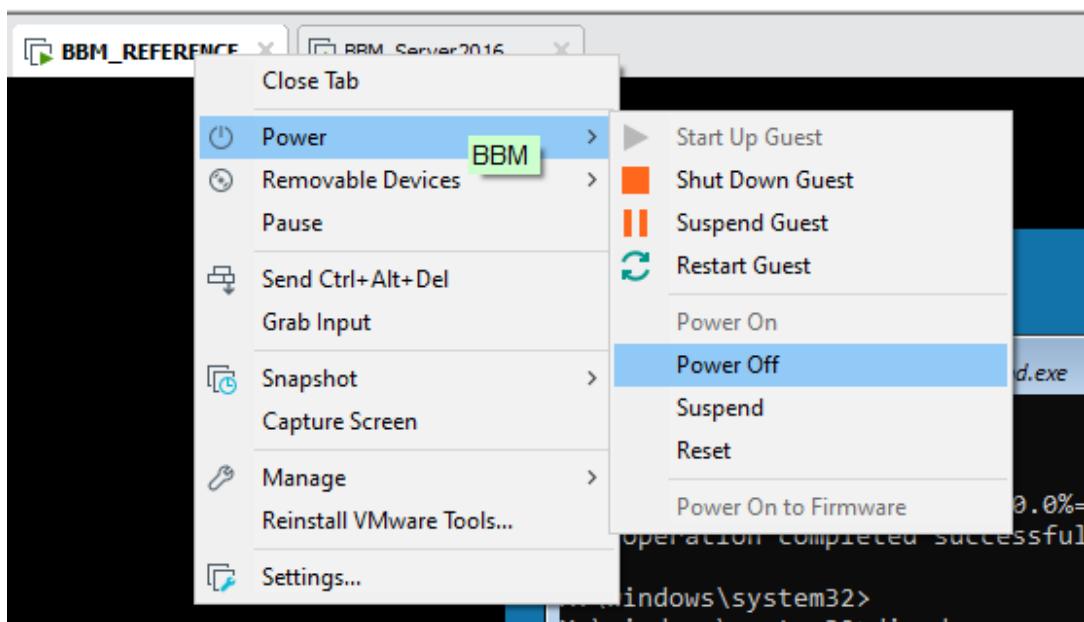
```
X:\windows\system32>
X:\windows\system32>dir d:
  Volume in drive D is BBM_UNNATTEND
  Volume Serial Number is B6BE-AF63

  Directory of D:\

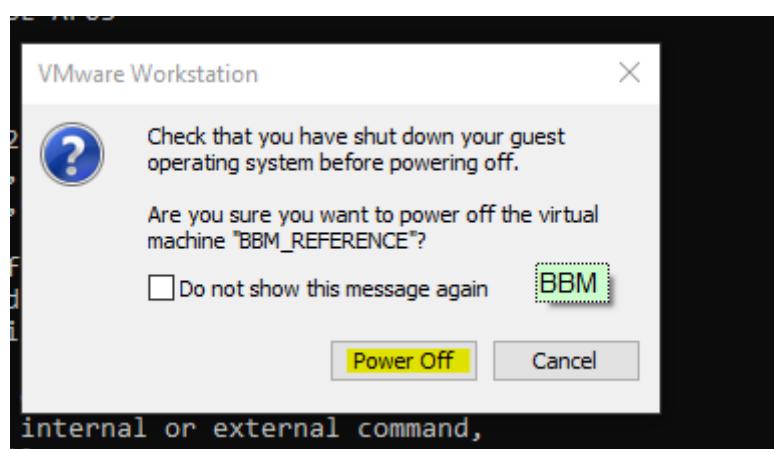
02/21/2021  09:05 PM      9,268,725,669 install.wim
              1 File(s)   9,268,725,669 bytes      BBM
              0 Dir(s)   11,634,737,152 bytes free

X:\windows\system32>
```

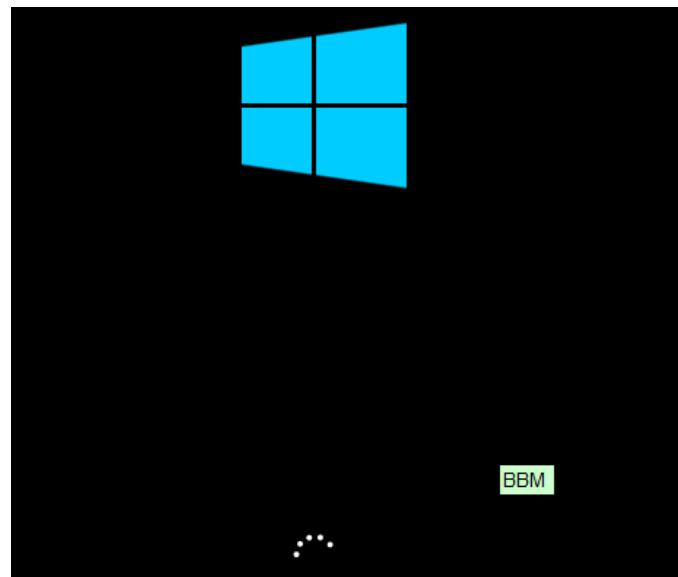
Ahora apagamos la máquina haciendo botón derecho sobre la pestaña de la máquina virtual > Power > Power off.



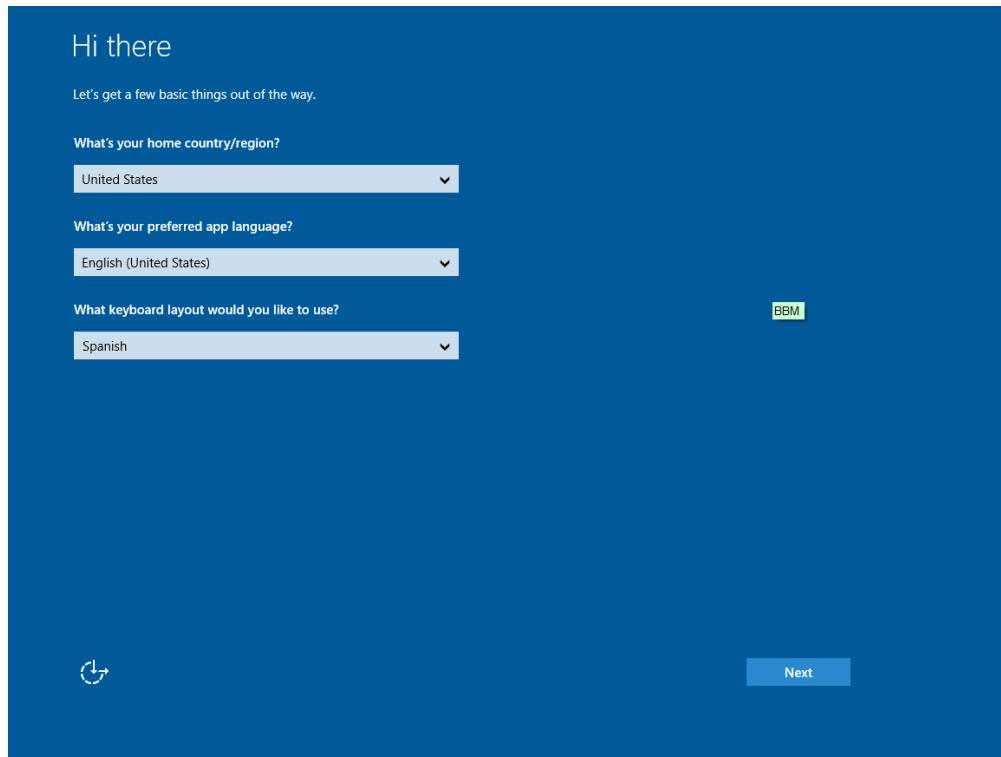
Sale una ventana en la que nos pregunta si realmente es lo que queremos hacer. Le damos a **Power off**.



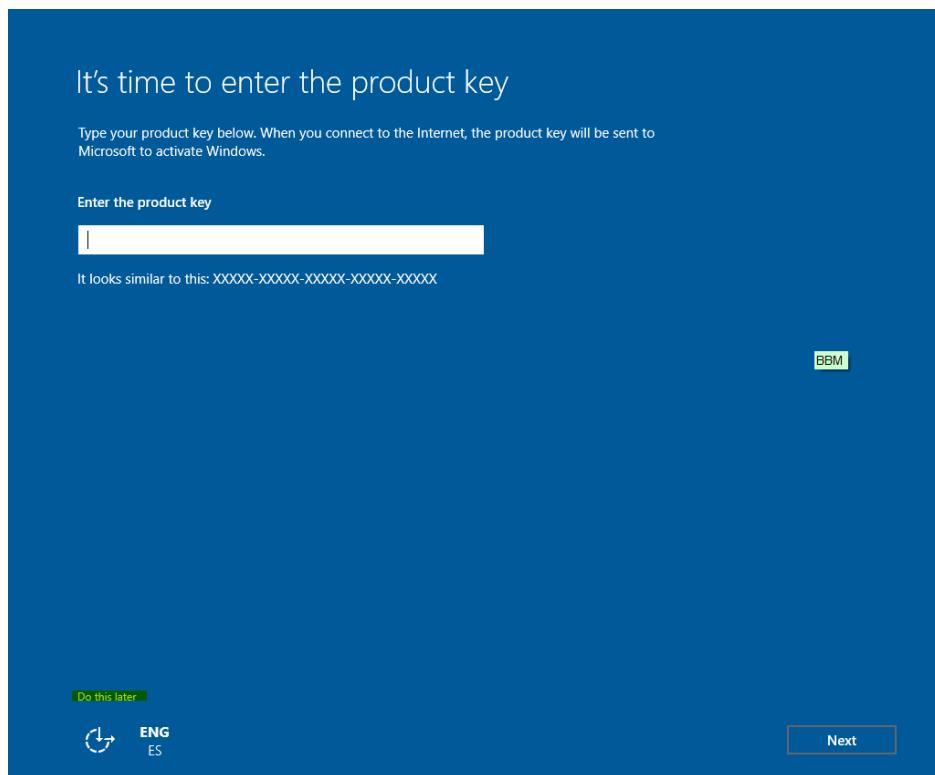
Iniciamos normal la máquina Technician:



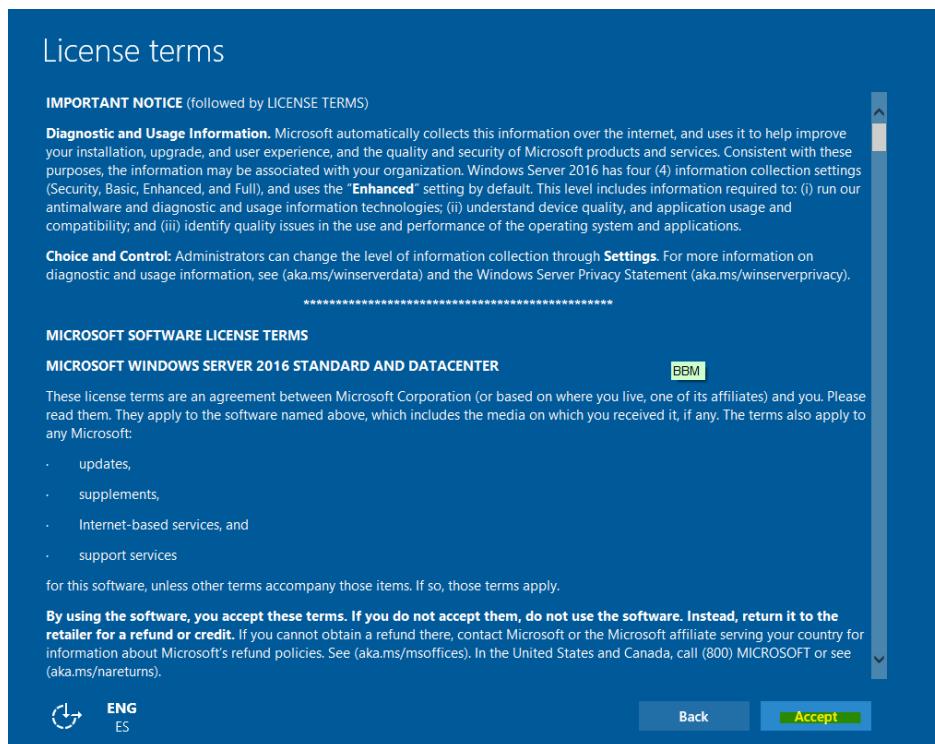
En la ventana del idioma dejamos lo que viene por defecto (United States). Esto es importante porque si no puede dar problemas más adelante. Lo único que podemos llegar a cambiar sin que afecte mucho, es el idioma del teclado, que podemos poner Spanish:



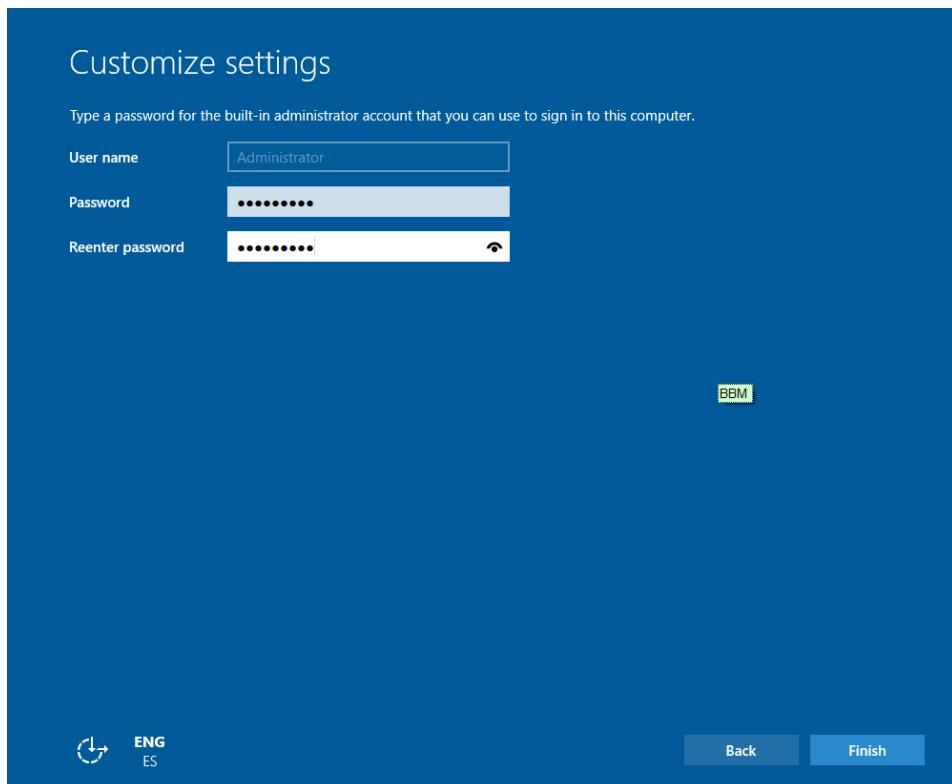
A la hora de meter el *Product Key*, le decimos que lo haremos más tarde (**Do this later**).



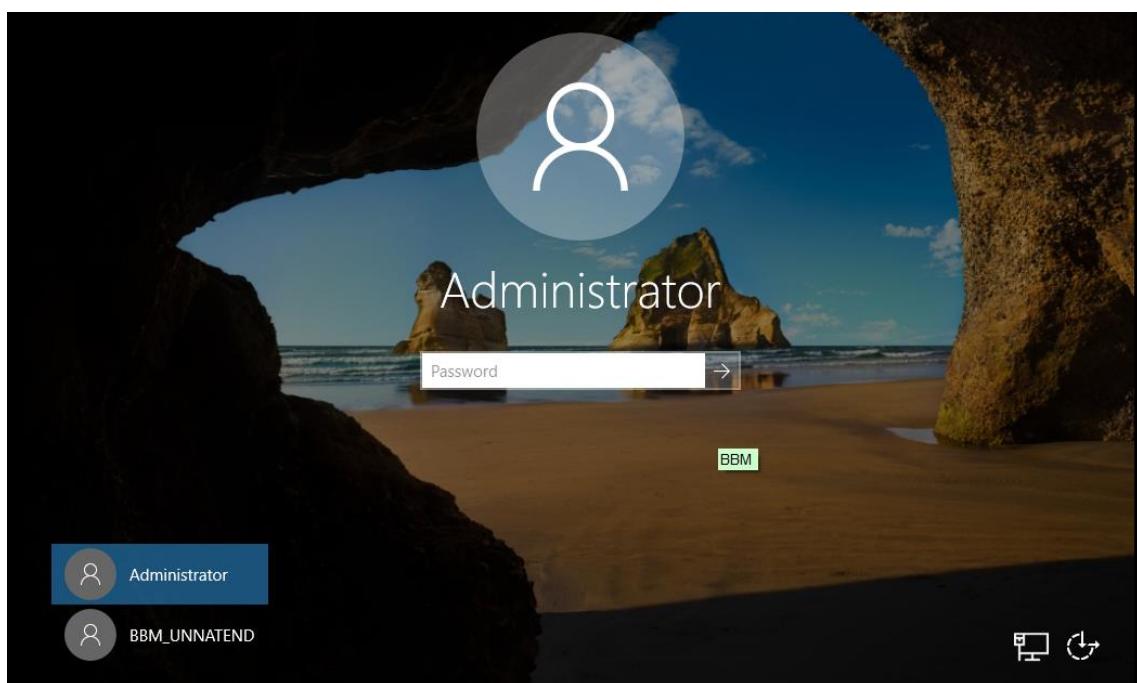
Aceptamos los términos de licencia:



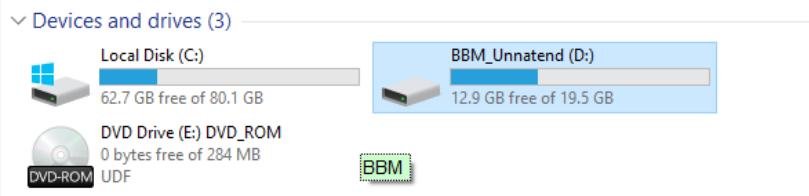
En la siguiente ventana le damos la contraseña (Abcd1234.)



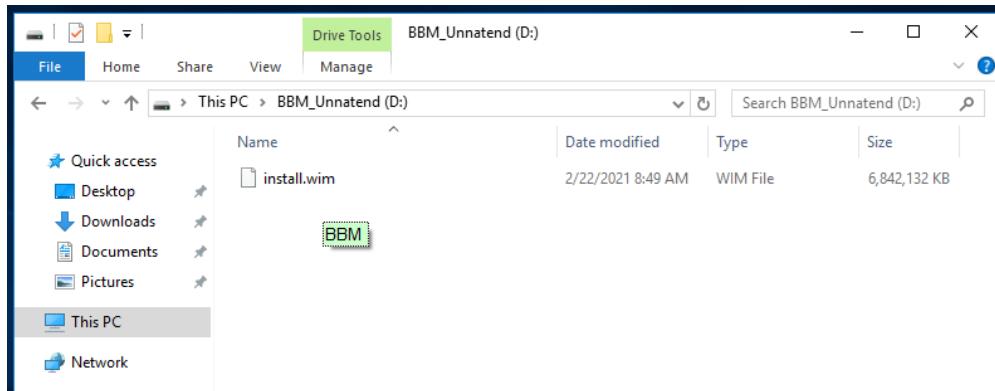
Con esto ya tenemos el inicio hecho (como Administrator):



Vemos que está iniciado con las particiones y dentro de la creada, tenemos un archivo **install.wim**.



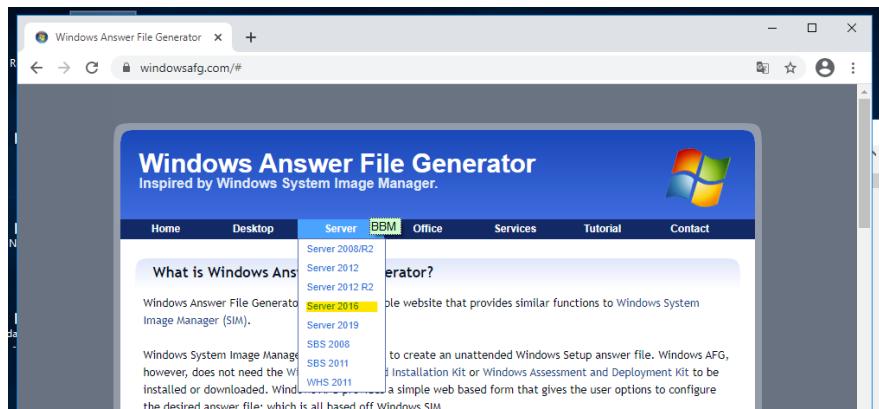
Vemos que dentro de la unidad D: está el archivo *install.wim*.



Lo copiamos a un pen y luego en la máquina **Tech**:

En la máquina **Tech** nos vamos a la página <https://www.windowsafg.com> (usar Firefox, que sino no se ve). Dado que el uso de una herramienta de terceros lleva mucho tiempo preparar la instalación desatendida de Windows 10, usar unattend.xml o autounattend.xml es una buena idea y también es mucho más fácil. Todo lo que tiene que hacer es generar un unattend.xml con los valores correctos y colocarlo en la raíz de su Windows 10 DVD o Windows 10 USB. Windows Answer File Generator es un servicio gratuito en línea para generar archivos.xml desatendidos para todas las versiones recientes de Windows, incluyendo la última versión de Windows 10. Usando este servicio gratuito, se puede generar un archivo de respuestas para Windows 10 Home, Windows 10 Pro, Windows 10 Education, Windows 10 Enterprise y Windows 10 Enterprise LTSB.

Una vez dentro de la página, se elige en este caso *Windows Server 2016* en el apartado *Server*.



El generador de archivos de respuestas de Windows 10 le permite seleccionar o configurar las siguientes cosas:

- Configuración general
 - Clave de producto
 - Aceptar el acuerdo de licencia
 - Activación automática
 - Configurar el idioma
 - Rearme de licencia
 - Nombre de la computadora
 - Nombre de la organización
- Configuración regional
 - Teclado o método de entrada
 - Moneda y formato de fecha
 - Zona horaria
 - Idioma de la interfaz de usuario
- Experiencia Fuera de la Caja (OOBE)
 - Ubicación de la red
 - Proteja su ordenador
 - Ocultar la página de EULA
 - Desactivar el ajuste automático del horario de luz diurna
 - Ocultar la configuración inalámbrica en OOBE
 - Vista del panel de control
 - Tamaño del ícono del panel de control
 - Saltar máquina OOBE
 - Saltar usuario OOBE
- Configuración de la partición
 - Disco de la esposa
 - Instalar en el disco
 - Partición principal activa
 - Formato de partición principal
 - Etiqueta de la partición principal
 - Carta de partición principal
 - Orden de partición
- Cuenta de usuario
 - Nombre de usuario
 - Grupo
 - Descripción
 - Contraseña (opcional)
 - Auto logon
 - UAC
 - Habilitar Firewall
 - CEIP

En Configuración de la partición, verá opciones para elegir varias configuraciones

NOTA: Dejo constancia del fichero entero capturado en la siguiente página

Windows Answer File Generator
Inspired by Windows System Image Manager.

Home Desktop Server Office Services Tutorial Contact

Windows Answer File Generator for Server 2016

Below are basic settings that can be configured during and after installation. What you type will automatically be updated in the "Output" area.

Installation Notes

Location:
Enter your comments here...

Notes:

General Settings

Product Key: WC2BQ-8NRM3-FDDYY-2BFGV-KHKQY
Accept EULA: Yes
Skip automatic activation: Yes
Skip License Rearm: Yes
Setup Language: English
Computer Name: BBM
Organization Name: BBM

Regional Settings

Keyboard or input method: English - United States
Currency and Date format: English - United States
Time Zone: (UTC+01:00) Brussels, Copenhagen, Madrid, Paris
UI Language: English - United States

Out Of Box Experience

Network Location: Work
Protect Your Computer: Recommended settings
Hide EULA page: Yes
Disable auto daylight timeset: No
Hide wireless setup in OOB: Yes
Skip machine OOB: Yes
Skip user OOB: Yes

Partition Settings

Wipe Disk: Yes
Install to disk: 0
Main Partition Active: Yes
Main Partition Format: NTFS
Main Partition Label: BBM_OS
Main Partition Letter: C
Partition Order: 2

User Account

User Name:
Group: Administrators
Description:
Password: (Optional)
Auto Logon: No
UAC: Disabled
Enable Firewall: Enable
CEIP: Disabled

Output

You may choose to create the autounattend.xml file automatically by clicking "Download File" link below, or doing it the manual way.

Ensure you select all the text below and copy to a word processor. Paste all the code and select "Save As...". Name the file "Autounattend.xml" without the quotations.

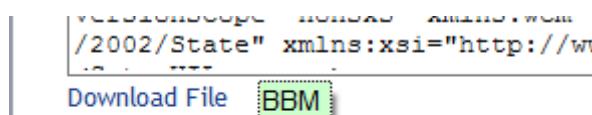
Place this file in the root of a USB drive and come back to a fresh install of Windows Server 2016.

```
<!--*****  
<!-*****  
Windows Server 2016 Answer File Generator  
Created using Windows AFG found at:  
;http://www.windowsafg.com  
*****-->  
*****  
*****  
<?xml version="1.0" encoding="utf-8"?>  
<unattend xmlns="urn:schemas-microsoft-com:unattend">  
<settings pass="windowsPE">  
<component name="Microsoft-Windows-International-Core-WinPE"  
processorArchitecture="x86" publicKeyToken="31bf3856ad364e35" language="neutral"  
versionScope="nonSxs" xmlns:wcm="http://schemas.microsoft.com/WMIConfig  
/2002/State" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">  
...  
Download File
```

Don't forget to follow us on Facebook.



Una vez cubierto el archivo, lo descargamos clicando Download File:



Don't forget to follow us on Facebook.

Una vez descargados hay que quitar las primeras líneas

```

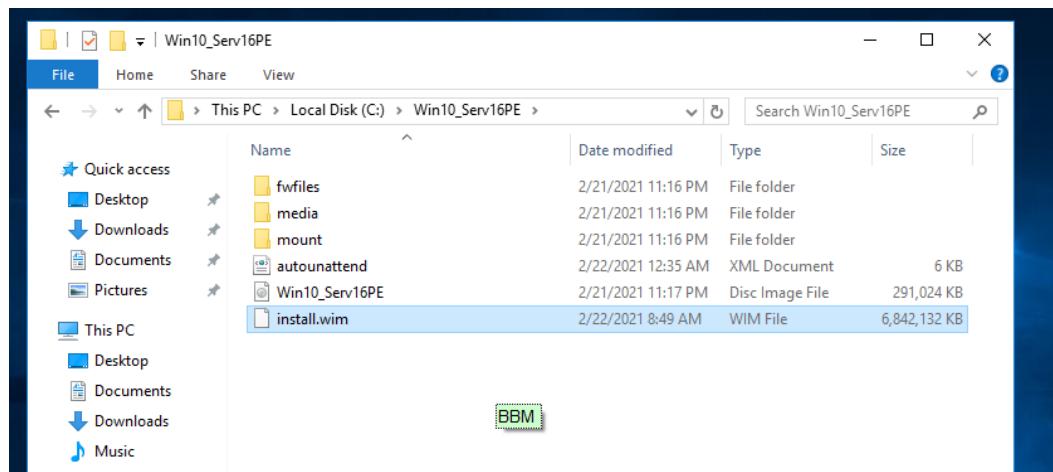
<!--*****-->
Windows Server 2016 Answer File Generator
Created using Windows AFG found at:
;http://www.windowsafg.com
BBM

Installation Notes
Location:
Notes: Enter your comments here...
*****-->

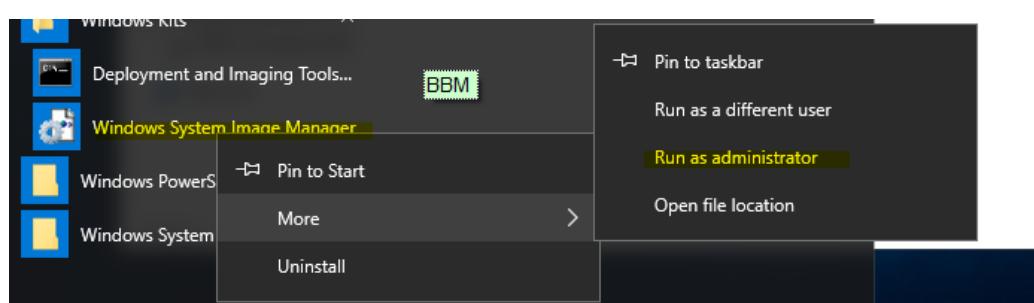
<?xml version="1.0" encoding="utf-8"?>
<unattend xmlns="urn:schemas-microsoft-com:unattend">
  <settings pass="windowsPE">
    <component name="Microsoft-Windows-International-Core-WinPE" processorArchitecture="amd64" publicKeyToken="31bf3856ad364e35" version="1.1" xmlns:wim="http://schemas.microsoft.com/wmi/>
      <SetupUILanguage>
        <UILanguage>en-US</UILanguage>

```

Ahora se copia en el directorio del **install.wim** en el directorio C:\Win10_Serv16PE



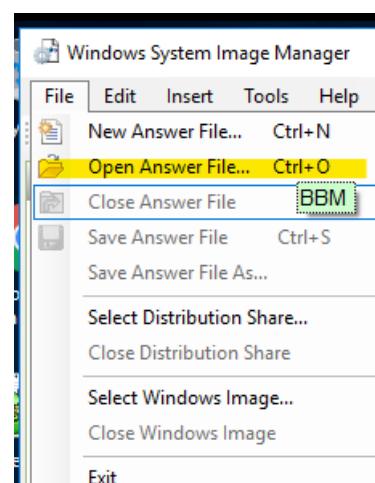
Abrimos como Administrador el Windows System Image Manager.



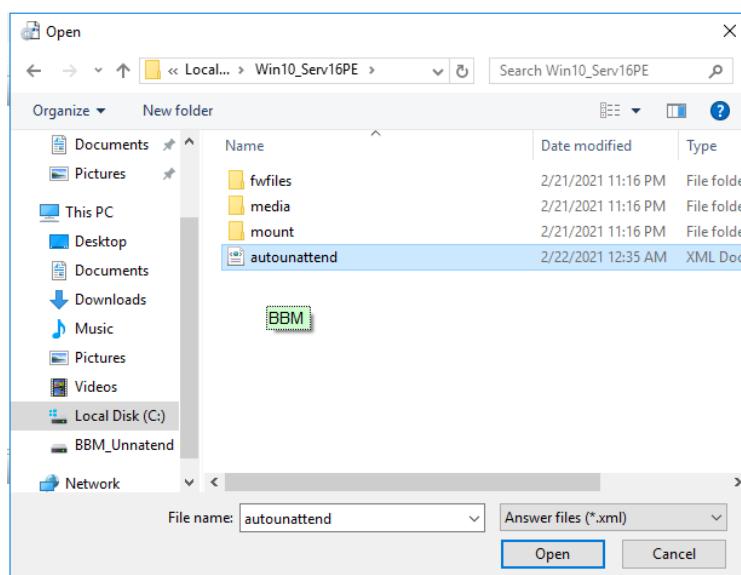
Aceptamos en el *pop-up* que se abre:



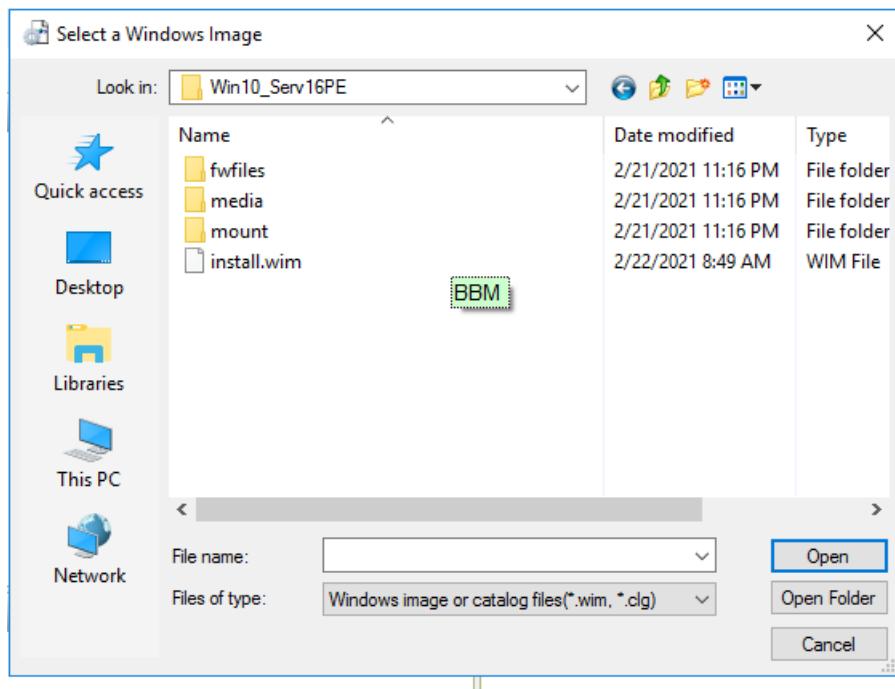
Ahora hay que cargar el xml. Para ello empleamos el **Windows System Image Manager**, este programa es la herramienta que utiliza para crear archivos de respuesta de instalación de Windows desatendidos. Para empezar hay que abrir el archivo de respuesta (Open Answer File):



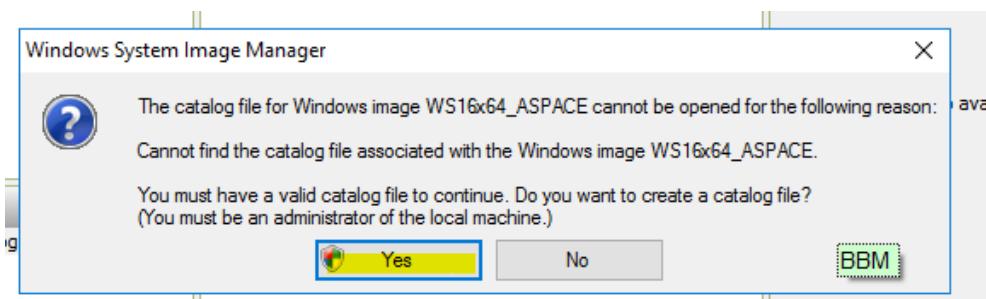
Elegimos el archivo **autounattend.xml** y le damos a **Open**.



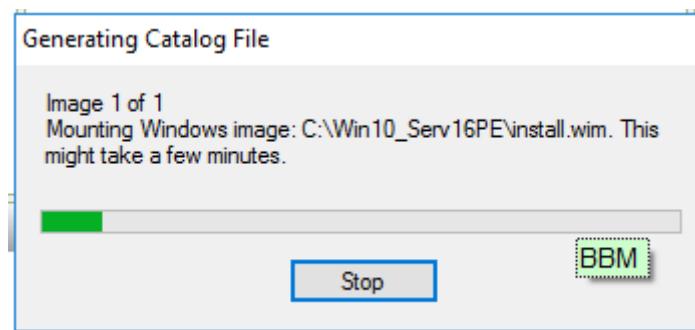
También abrimos el install.wim:



Cuando nos sale el *pop-up* de advertencia, le damos a **yes**. Esto solo sale la primera vez.



Nos sale una ventana de carga. Con esto ya quedaría todo cargado:



Ahora ya quedaría configurar los componentes (**Components**). Hay varias opciones. Lo que hace en las opciones Es la configuración que se ve en las imágenes. Vamos por pasos:

En la parte 1 Windows PE, se puede configurar el lenguaje

Answer File

- autounattend
 - Components
 - 1 windowsPE
 - amd64_Microsoft-Windows-International-Core-WinPE_neutral
 - SetupUILanguage

SetupUILanguage Properties

Properties	AppliedConfigurationPass	1 windowsPE
Component	Microsoft-Windows-International-Core-WinF	SetupUILanguage
Path		
Settings	UILanguage	en-US
	WillShowUI	OnError
		Always
		OnError
		Never

Answer File

- autounattend
 - Components
 - 1 windowsPE
 - amd64_Microsoft-Windows-International-Core-WinPE_neutral
 - DiskConfiguration
 - ImageInstall
 - UserData

Microsoft-Windows-Setup Properties

Properties	AppliedConfigurationPass	1 windowsPE
Enabled	True	
> Id	amd64_Microsoft-Windows-Setup_neutral	
Settings	EnableFirewall	true
	EnableNetwork	true
	LogPath	true
	Restart	false
	UseConfigurationSet	

También la configuración de los discos

Answer File

- autounattend
 - Components
 - 1 windowsPE
 - amd64_Microsoft-Windows-International-Core-WinPE_neutral
 - DiskConfiguration
 - Disk[DiskID="0"]
 - CreatePartitions
 - ModifyPartitions
 - ImageInstall
 - UserData

Disk[DiskID="0"] Properties

Properties	AppliedConfigurationPass	1 windowsPE
Component	Microsoft-Windows-Setup	
KeyName	DiskID	
Path	DiskConfiguration/Disk[DiskID="0"]	
Settings	Action	AddList Item
	DiskID	0
	WillWipeDisk	true

Answer File

- autounattend
 - Components
 - 1 windowsPE
 - amd64_Microsoft-Windows-International-Core-WinPE_neutral
 - DiskConfiguration
 - Disk[DiskID="0"]
 - CreatePartitions
 - CreatePartition[Order="1"]
 - CreatePartition[Order="2"]
 - ModifyPartitions

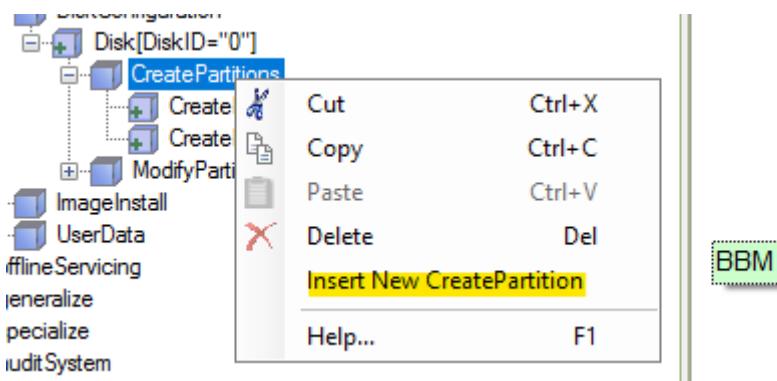
CreatePartition[Order="1"] Properties

Properties	AppliedConfigurationPass	1 windowsPE
Component	Microsoft-Windows-Setup	
KeyName	Order	
Path	DiskConfiguration/Disk[DiskID="0"]/CreatePartition[Order="1"]	
Settings	Action	AddList Item
	Extend	false
	Order	1
	Size	300
	Type	Primary

CreatePartition[Order="2"] Properties

- Properties**
 - AppliedConfigurationPass: 1 windowsPE
 - Component: Microsoft-Windows-Setup
 - KeyName:
 - Path: DiskConfiguration/Disk[DiskID="0"]/Create
- Settings**
 - Action: AddList Item
 - Extend: false
 - Order: 2
 - Size: 200
 - Type: **EFI** (highlighted)

Como quiero crear dos particiones más, hacemos botón derecho sobre *Create Partition* y le damos a *Insert New CreatePartition*.



Una de las particiones (la tres) será de tipo **partición reservada de Microsoft (MSR)**, que es una partición de un dispositivo de almacenamiento de datos, que se crea para reservar una parte del espacio en disco para un posible uso posterior por parte de un sistema operativo Windows instalado en una partición separada. No se almacenan datos significativos dentro del MSR; aunque del MSR, se pueden tomar fragmentos para la creación de nuevas particiones, que a su vez pueden contener estructuras de datos.

CreatePartition[Order="3"] Properties

- Properties**
 - AppliedConfigurationPass: 1 windowsPE
 - Component: Microsoft-Windows-Setup
 - KeyName:
 - Path: DiskConfiguration/Disk[DiskID="0"]/Create
- Settings**
 - Action: AddList Item
 - Extend: false
 - Order: 3
 - Size: 128
 - Type: **MSR** (highlighted)

La última es el disco que se va a ver

Answer File

CreatePartition[Order="4"] Properties	
Properties	
AppliedConfigurationPass	1 windowsPE
Component	Microsoft-Windows-Setup
KeyName	Order
Path	DiskConfiguration/Disk[DiskID="0"]/CreatePartitions/CreatePartition[Order="4"]
Settings	
Action	AddList Item
Extend	false
Order	4
Size	60000
Type	Primary

Ahora se modifican los archivos

Answer File

ModifyPartition[Order="1"] Properties	
Properties	
AppliedConfigurationPass	1 windowsPE
Component	Microsoft-Windows-Setup
KeyName	Order
Path	DiskConfiguration/Disk[DiskID="0"]/ModifyPartitions/ModifyPartition[Order="1"]
Settings	
Action	AddList Item
Active	true
Extend	false
Format	NTFS
Label	WINRE
Letter	1
Order	1
PartitionID	1
TypeID	DE94BBA4-06D1-4D40-A16A-BFD50179D6AC

Answer File

ModifyPartition[Order="2"] Properties	
Properties	
AppliedConfigurationPass	1 windowsPE
Component	Microsoft-Windows-Setup
KeyName	Order
Path	DiskConfiguration/Disk[DiskID="0"]/ModifyPartitions/ModifyPartition[Order="2"]
Settings	
Action	AddList Item
Active	true
Extend	false
Format	FAT32
Label	System
Letter	2
Order	2
PartitionID	2
TypeID	0x0000

Answer File

ModifyPartition[Order="3"] Properties

Properties	AppliedConfigurationPass Component KeyName Path	1 windowsPE Microsoft-Windows-Setup Order DiskConfiguration/Disk[DiskID="0"]/ModifyPartitions/Mo
Settings	Action Active Extend Format Label Letter Order PartitionID TypeID	AddList Item 3 BBM 3

Answer File

ModifyPartition Properties

Properties	AppliedConfigurationPass Component KeyName Path	1 windowsPE Microsoft-Windows-Setup Order DiskConfiguration/Disk[DiskID="0"]/ModifyPartitions/Mo
Settings	Action Active Extend Format Label Letter Order PartitionID TypeID	AddList Item NTFS BBM_OS 4 4 BBM

NOTA: Donde se va a instalar el sistema operativo en el volumen 0 partición 4

Ahora configuraremos el archivo de imagen de instalación.

Answer File

InstallTo Properties

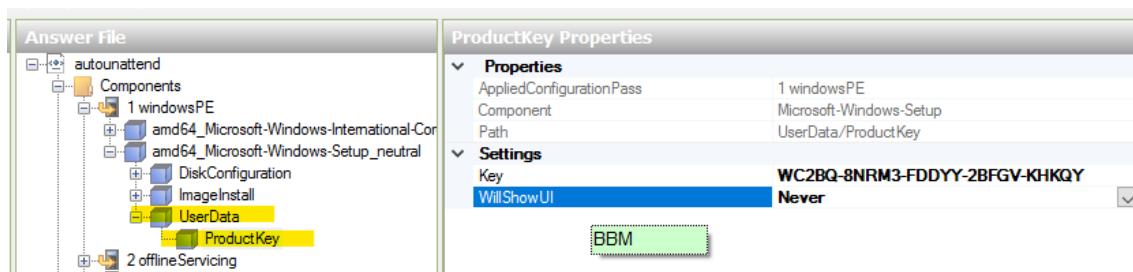
Properties	AppliedConfigurationPass Component Path	1 windowsPE Microsoft-Windows-Setup ImagelInstall/OSImage/InstallTo
Settings	DiskID PartitionID	0 4 BBM

También en este apartado configuraremos los datos de usuario

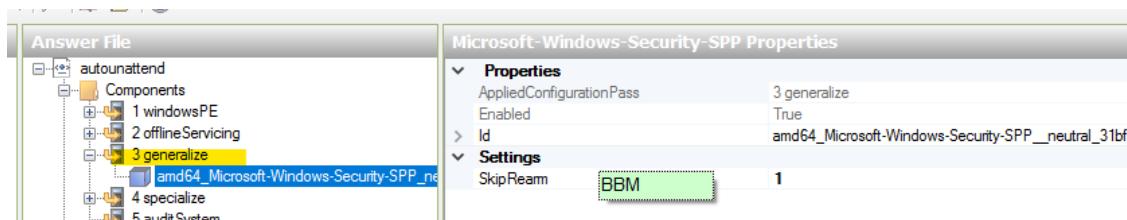
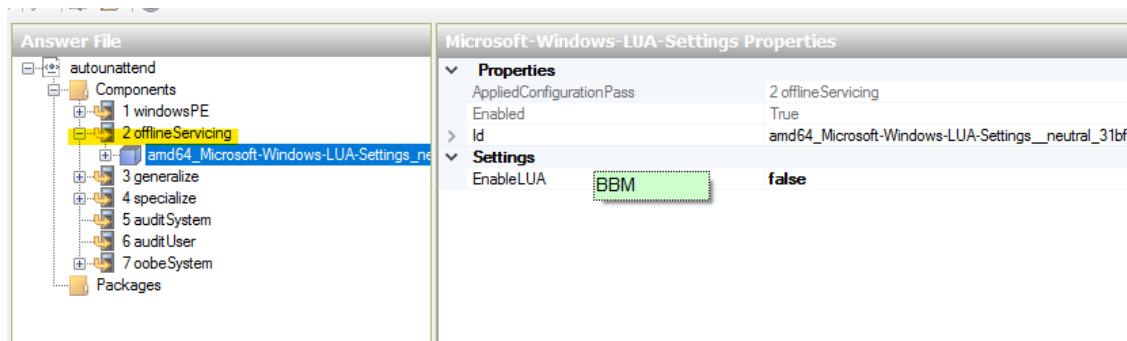
Answer File

UserData Properties

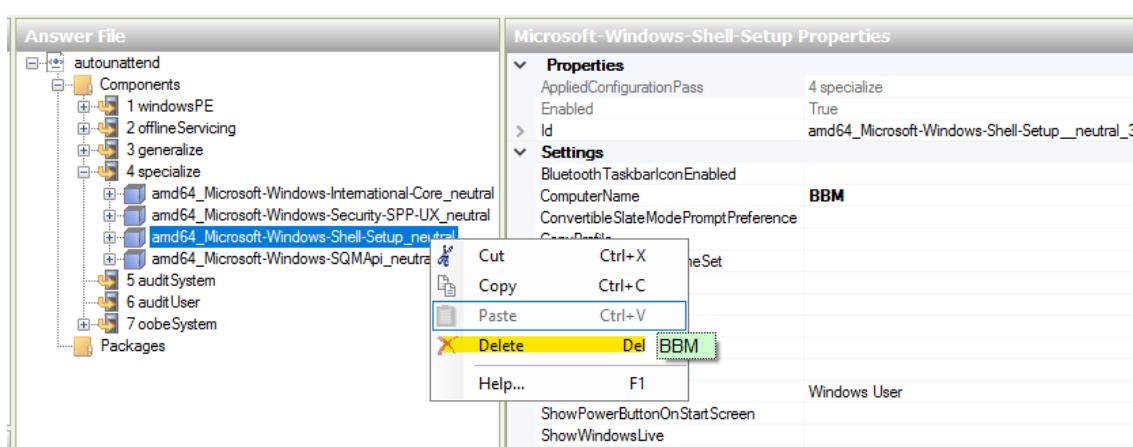
Properties	AppliedConfigurationPass Component Path	1 windowsPE Microsoft-Windows-Setup UserData
Settings	AcceptEula FullName Organization	true BBM BBM



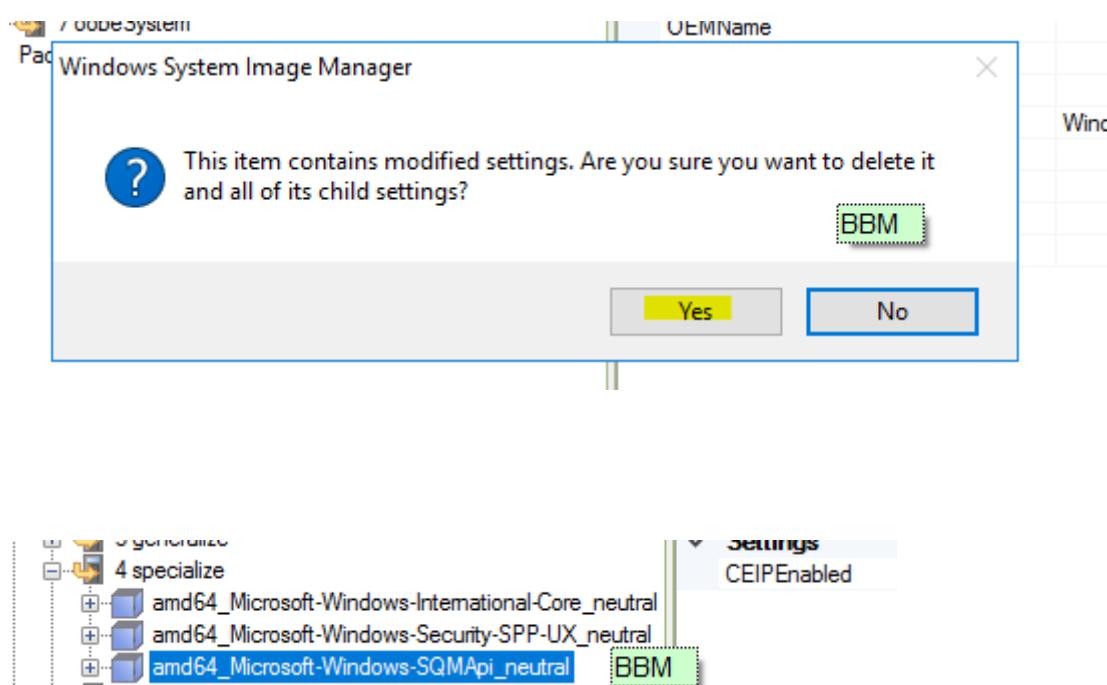
En el apartado 2 office Servicing y en el 3 generalize se dejan por defecto.



Con este apartado 4 specialize la instalación desatendida falla. Por eso se borra. Se le da a botón derecho y le da a Delete.



Aceptamos el Delete.

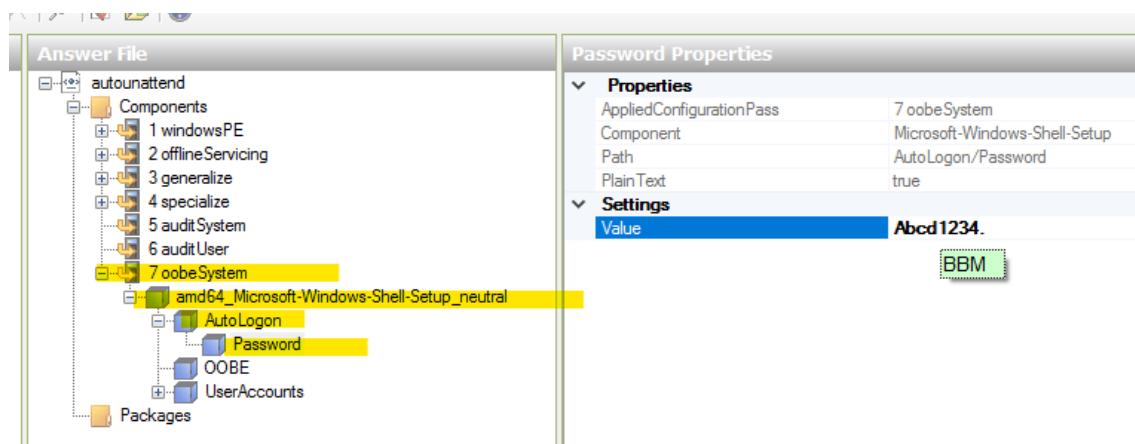


El 5 audit System y 6 audit System. En el 7 oobe System hacemos lo siguiente

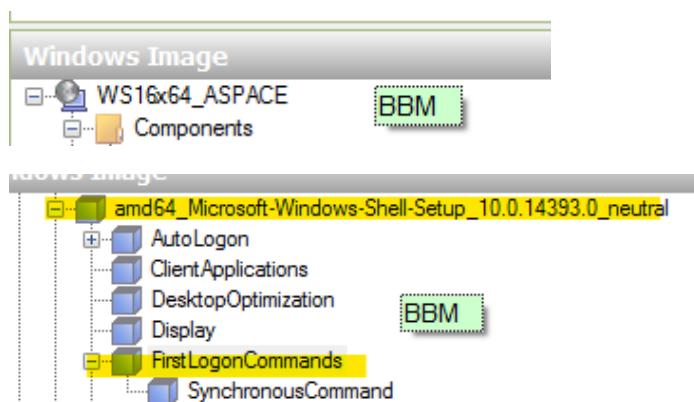
The screenshot shows the 'Microsoft-Windows-Shell-Setup Properties' window in the Answer File editor. The 'Properties' section shows 'AppliedConfigurationPass' as '7 oobeSystem' and 'Enabled' as 'True'. The 'Settings' section shows 'BluetoothTaskbarIconEnabled' as 'false', 'ConvertibleSlateModePromptPreference' as 'true', 'DisableAutoDaylightTimeSet' as 'false', 'DoNotCleanTaskBar' as 'true', 'EnableStartMenu' as 'true', 'RegisteredOrganization' as 'BBM', and 'RegisteredOwner' as 'BBM'. A green dashed box highlights the 'BBM' label next to the 'RegisteredOwner' key.

The screenshot shows the 'AutoLogon Properties' window in the Answer File editor. The 'Properties' section shows 'AppliedConfigurationPass' as '7 oobeSystem', 'Component' as 'Microsoft-Windows-Shell-Setup', and 'Path' as 'AutoLogon'. The 'Settings' section shows 'Domain' as 'localhost', 'Enabled' as 'false', and 'LogonCount' as '1'. A green dashed box highlights the 'BBM' label next to the 'LogonCount' key.

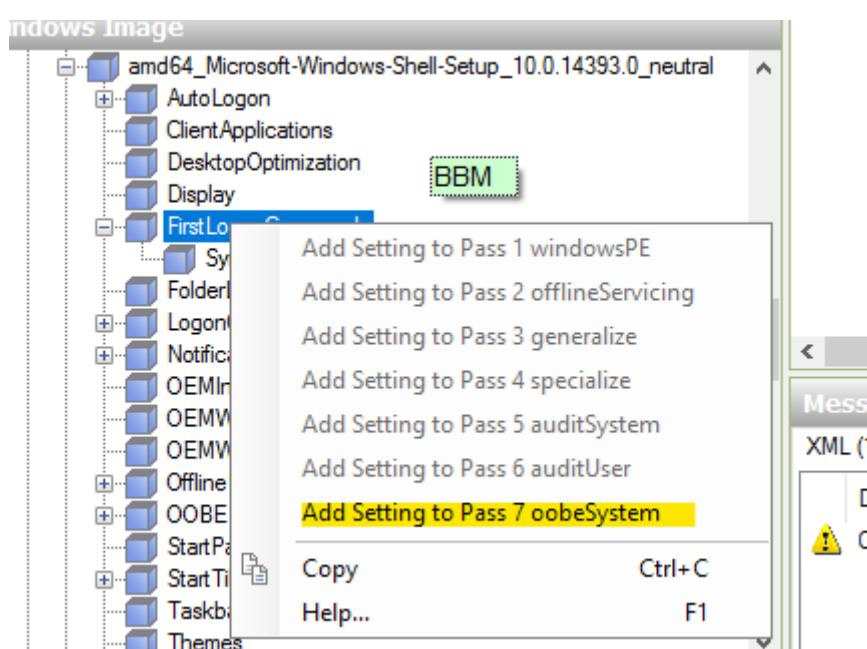
En el apartado del 7 oobe System, tenemos que agregar los First Logon commands, que son tres registros que nos ayudarán en la instalación.



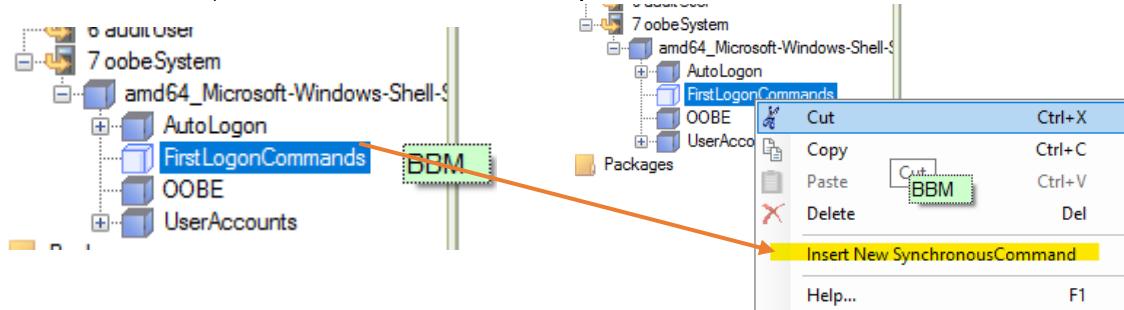
Casualmente coincide con lo que borré en 3 generalize. Para agregarlo voy a la parte derecha del programa y busco en Windows Image, en el componente *amd64_Microsoft-Windows-Shell-Setup_10.0.14393.0_neutral*. Clico en FirstLogonCommands.



Ahí le damos con el botón derecho y le damos a Add Setting to Pass 7 oobe System.

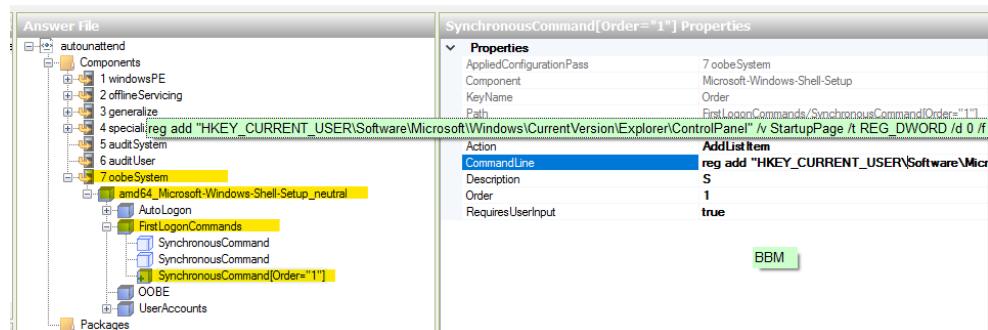


Como se ve en la imagen, queda agregado a la opción 7. Ahora le damos con el botón derecho y le damos a **Insert New Synchronous Command**.

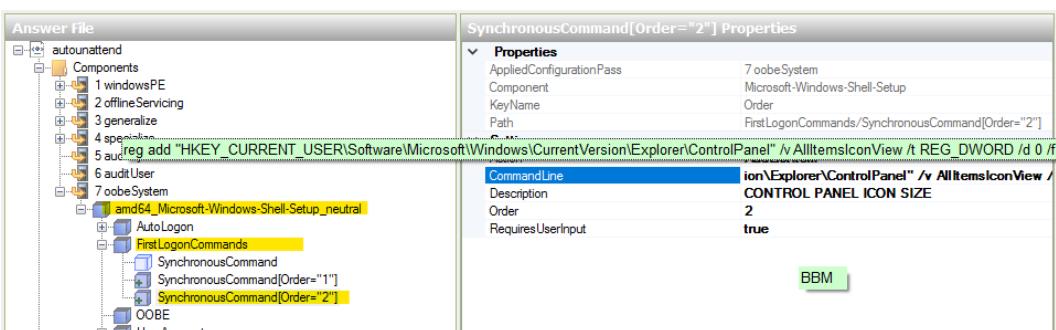


Introducimos tres líneas que se añaden a **regedit**.

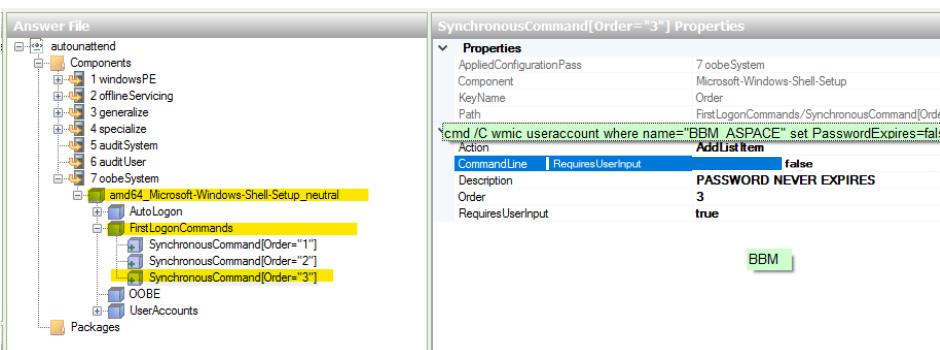
```
reg add
"HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\ControlPanel" /v StartupPage /t REG_DWORD /d 0 /f
```



```
reg
add
"HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\ControlPanel" /v AllItemsIconView /t REG_DWORD /d 0 /f
```



```
cmd /C wmic useraccount where name="BBM_ASPACE" set PasswordExpires=false
```



Ahora continuamos con la configuración:

The screenshot shows the WinPE Answer File Editor interface. On the left, the tree view shows the 'Answer File' structure under 'autounattend'. The '7 oobeSystem' node is selected. On the right, the 'OOBE Properties' section is displayed with the following settings:

Setting	Value
HideEULAPage	true
HideLocalAccountScreen	true
HideOEMRegistrationScreen	true
HideOnlineAccountScreens	true
HideWirelessSetupInOOBE	true
NetworkLocation	Work
OEMAppld	3
ProtectYourPC	false
SkipMachineOOBE	false
SkipUserOOBE	false
UnattendEnableRetailDemo	

En User accounts le metemos al Admin, con la contraseña (Abcd1234.)

The screenshot shows the WinPE Answer File Editor interface. The 'Answer File' tree view shows the 'UserAccounts' node under '7 oobeSystem'. On the right, the 'AdministratorPassword Properties' section is displayed with the following setting:

Setting	Value
Value	Abcd1234.

Meto también una cuenta local secundaria

The screenshot shows the WinPE Answer File Editor interface. The 'Answer File' tree view shows the 'LocalAccounts' node under 'UserAccounts'. On the right, the 'LocalAccount[Name=""'] Properties' section is displayed with the following settings:

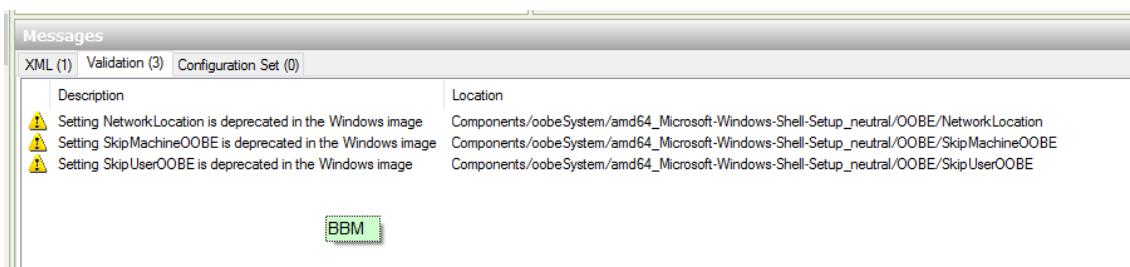
Setting	Value
Action	AddList Item
Description	
DisplayName	
Group	Administrators
Name	BBM_Unnated

Below this, there is a note: 'Name Type: String, MaxLength: 20'.

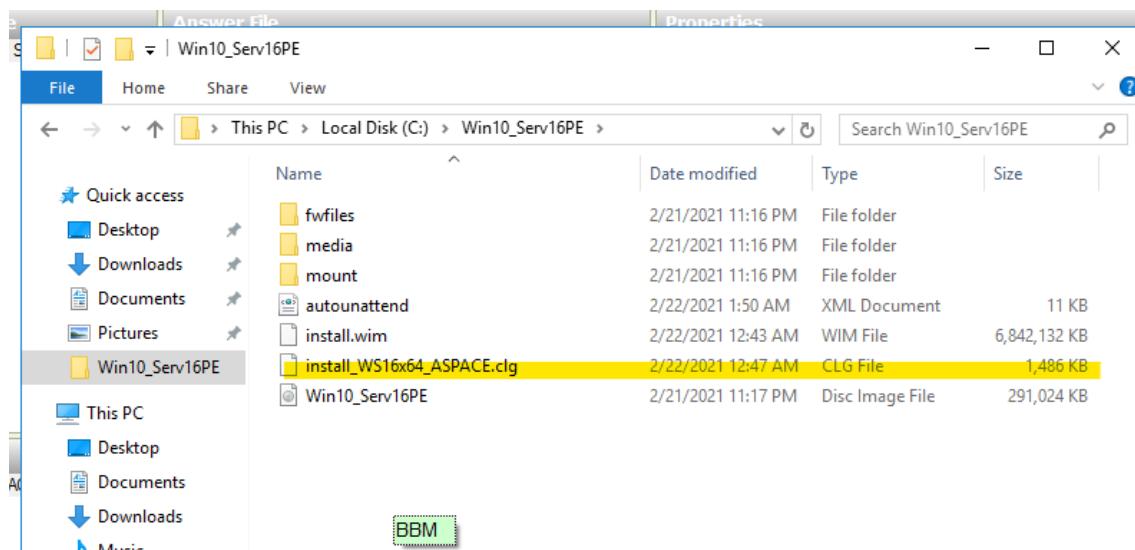
Al acabar esto, le doy a Validate Answer File.



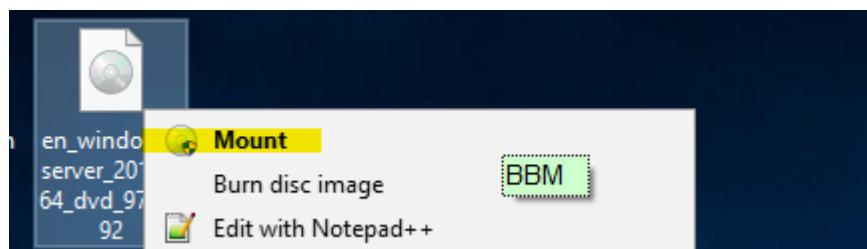
Salen unos mensajes de advertencia, que están *deprecated* desde la versión de Windows 7.



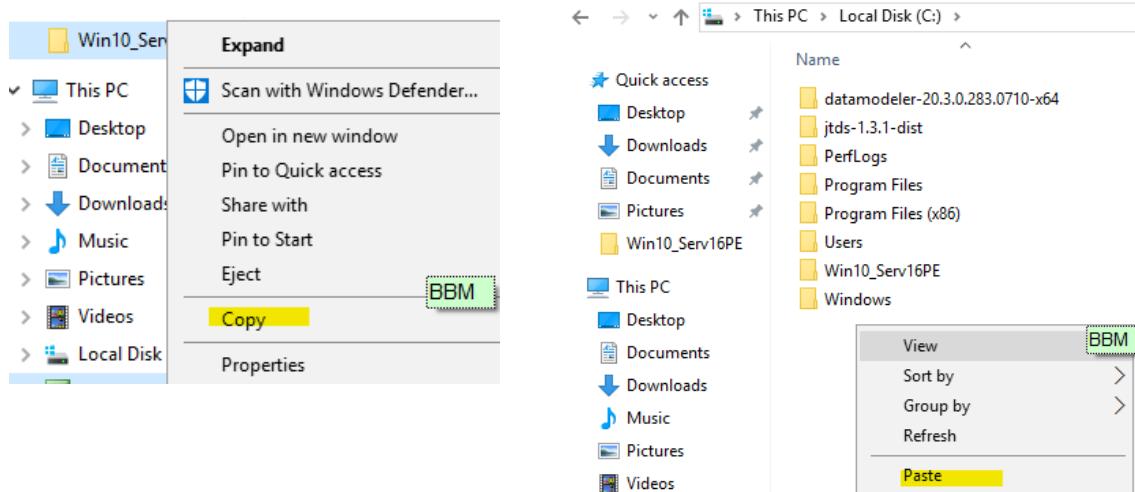
Le doy a **Save** y se guarda la plantilla del archivo de respuestas.



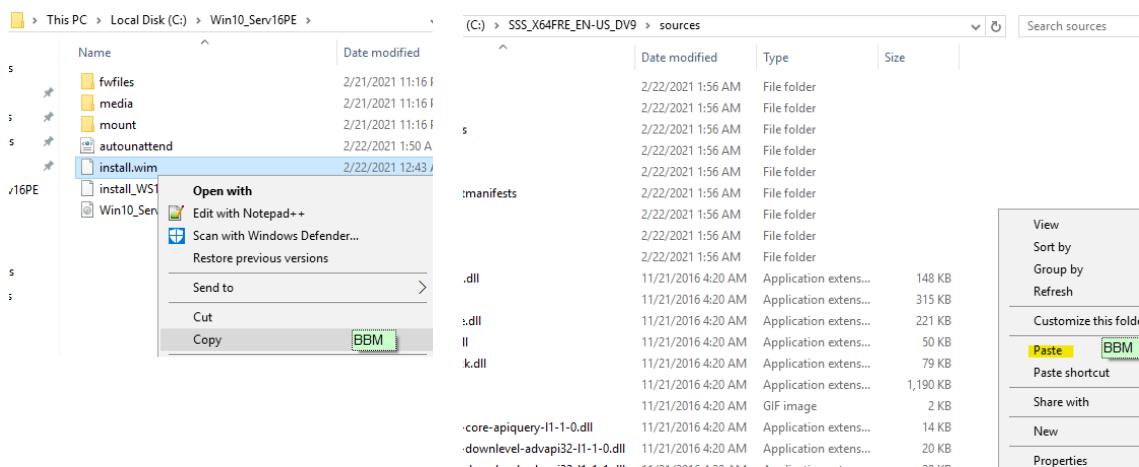
Ahora cogemos una ISO, y la montamos.



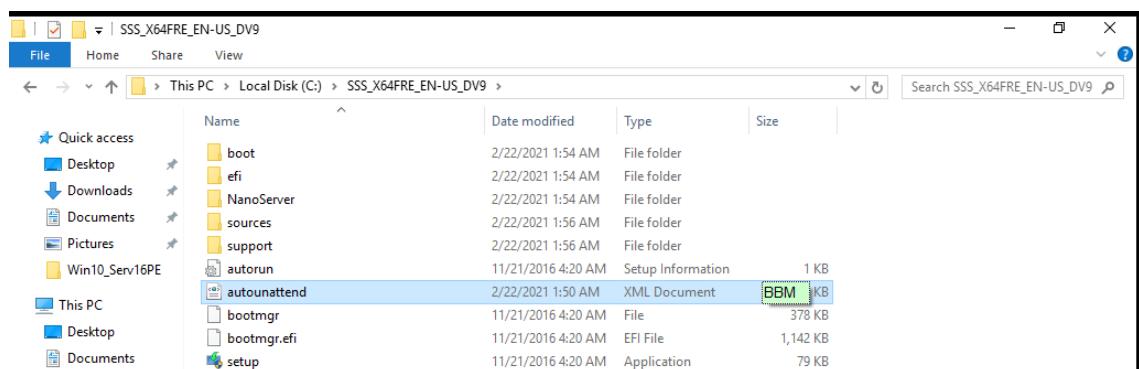
Ahora hago **Copy-Paste** en la raíz



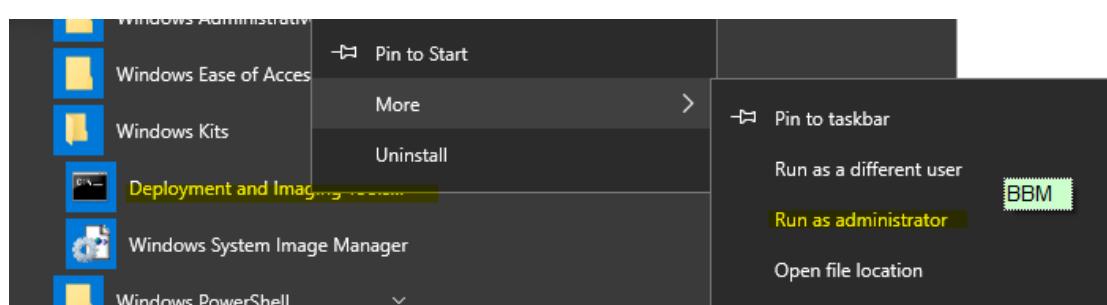
Ahora entramos en la carpeta donde teníamos el `install.wim` y lo cambiamos por el nuestro.



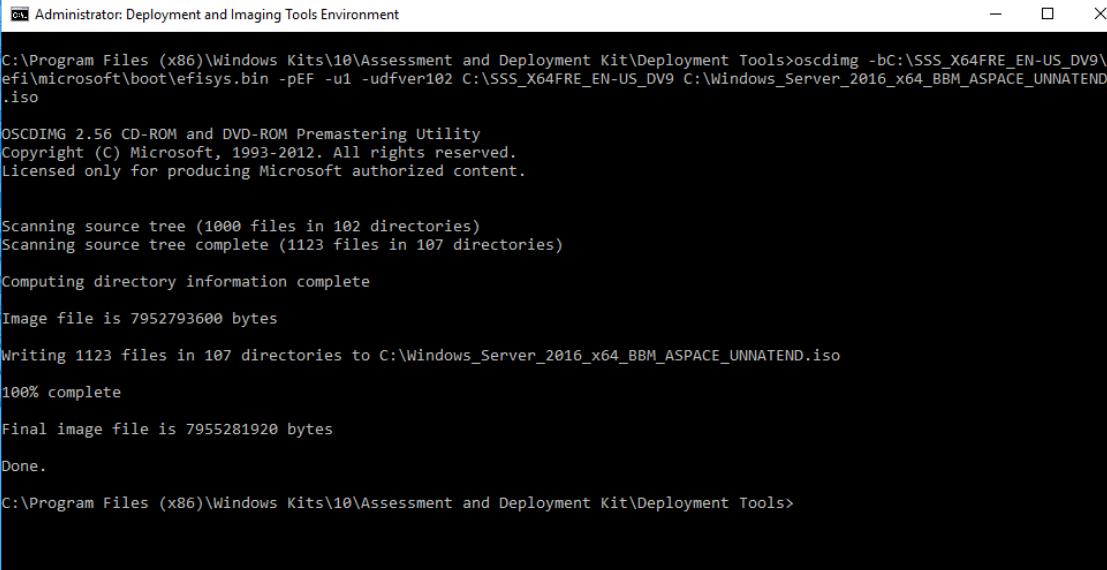
Hacemos lo mismo con el `autounattend.xml`.



Volvemos al Deployment and Imaging Tools Environment



Escribimos el siguiente comando: `oscdimg -bC:\SSS_X64FRE_EN-US_DV9\efi\microsoft\boot\efisys.bin -pEF -u1 -udfver102 C:\SSS_X64FRE_EN-US_DV9 C:\Windows_Server_2016_x64_BBM_ASPACE_UNNATEND.iso.`



```
C:\> Administrator: Deployment and Imaging Tools Environment
C:\Program Files (x86)\Windows Kits\10\Assessment and Deployment Kit\Deployment Tools>oscdimg -bC:\SSS_X64FRE_EN-US_DV9\efi\microsoft\boot\efisys.bin -pEF -u1 -udfver102 C:\SSS_X64FRE_EN-US_DV9 C:\Windows_Server_2016_x64_BBM_ASPACE_UNNATEND.iso

OSCDIMG 2.56 CD-ROM and DVD-ROM Premastering Utility
Copyright (C) Microsoft, 1993-2012. All rights reserved.
Licensed only for producing Microsoft authorized content.

Scanning source tree (1000 files in 102 directories)
Scanning source tree complete (1123 files in 107 directories)

Computing directory information complete

Image file is 7952793600 bytes

Writing 1123 files in 107 directories to C:\Windows_Server_2016_x64_BBM_ASPACE_UNNATEND.iso

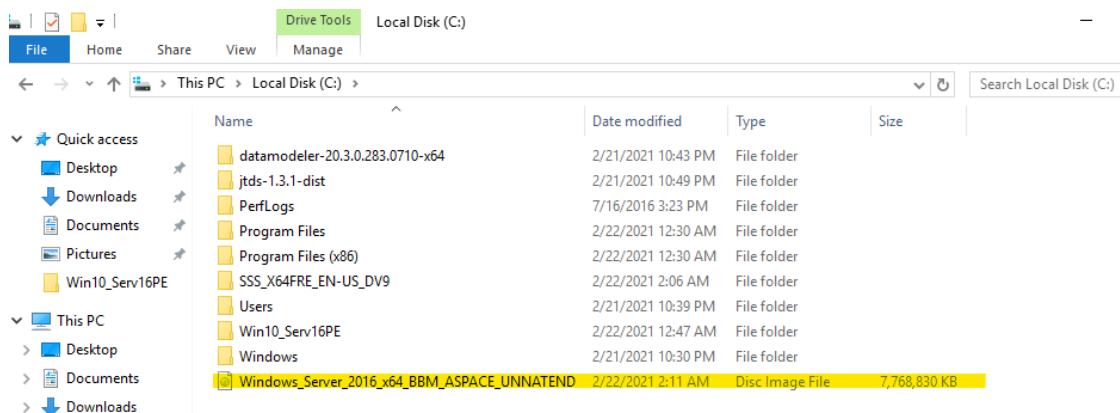
100% complete

Final image file is 7955281920 bytes

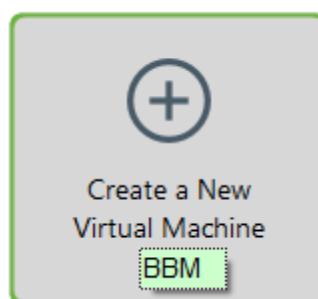
Done.

C:\Program Files (x86)\Windows Kits\10\Assessment and Deployment Kit\Deployment Tools>
```

Ahora ya tenemos una ISO preparada para hacer una instalación desatendida.



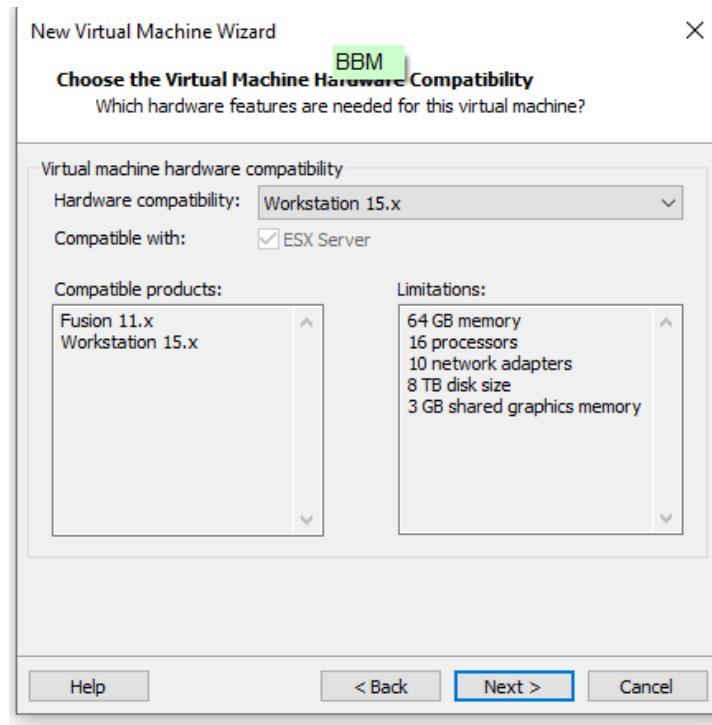
El siguiente paso es crear una Máquina Virtual Nueva. Le damos sobre el botón **Create a New Virtual Machine**.



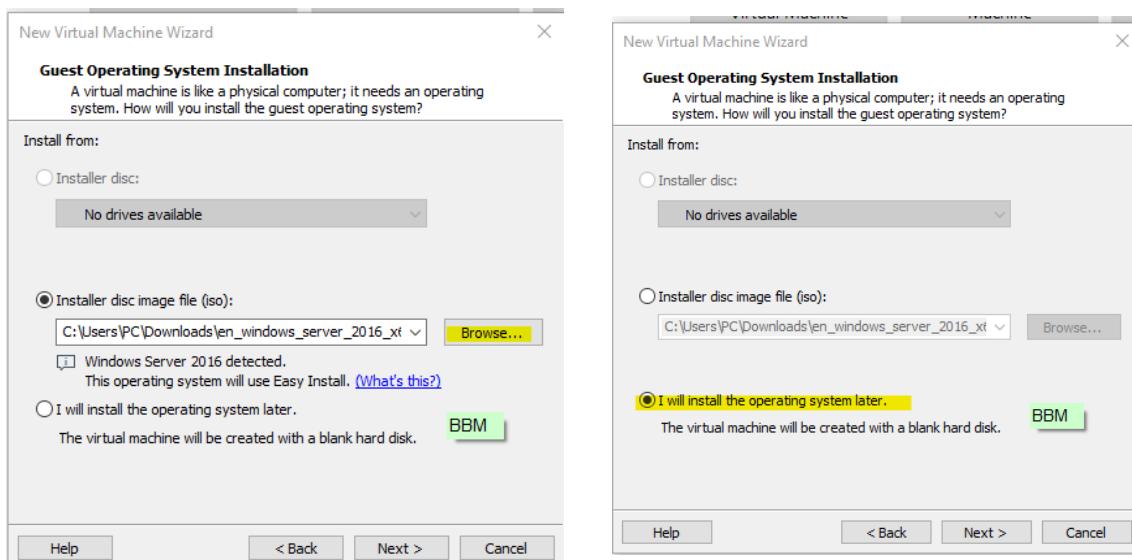
En la primera página elegimos instalación *Custom* porque queremos dejarla a nuestro gusto por completo:



La siguiente página es por defecto:



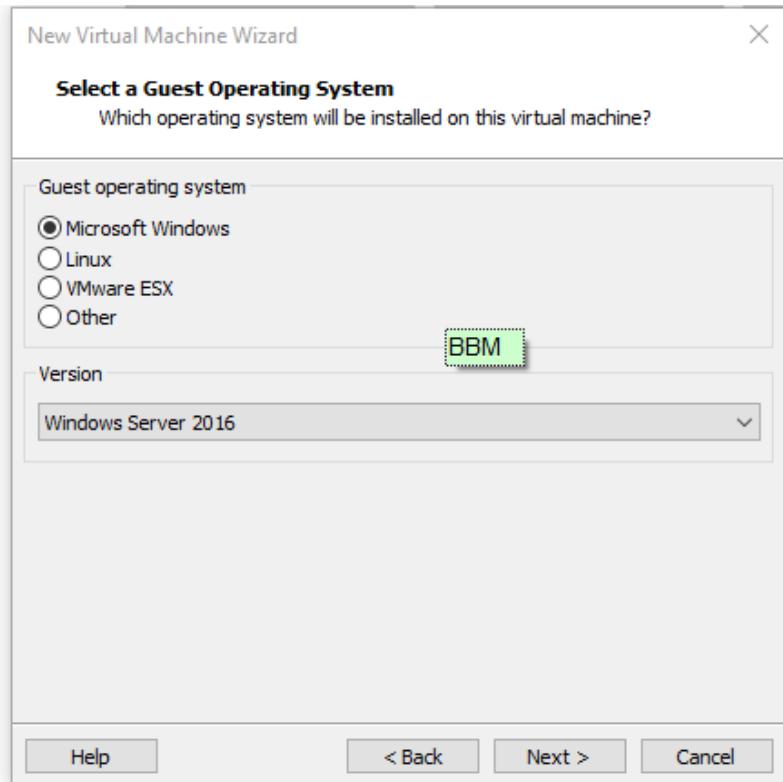
En la siguiente como Installer disc image file (iso) elegimos la Imagen que hemos creado y luego le damos a I Will install the operating system later.



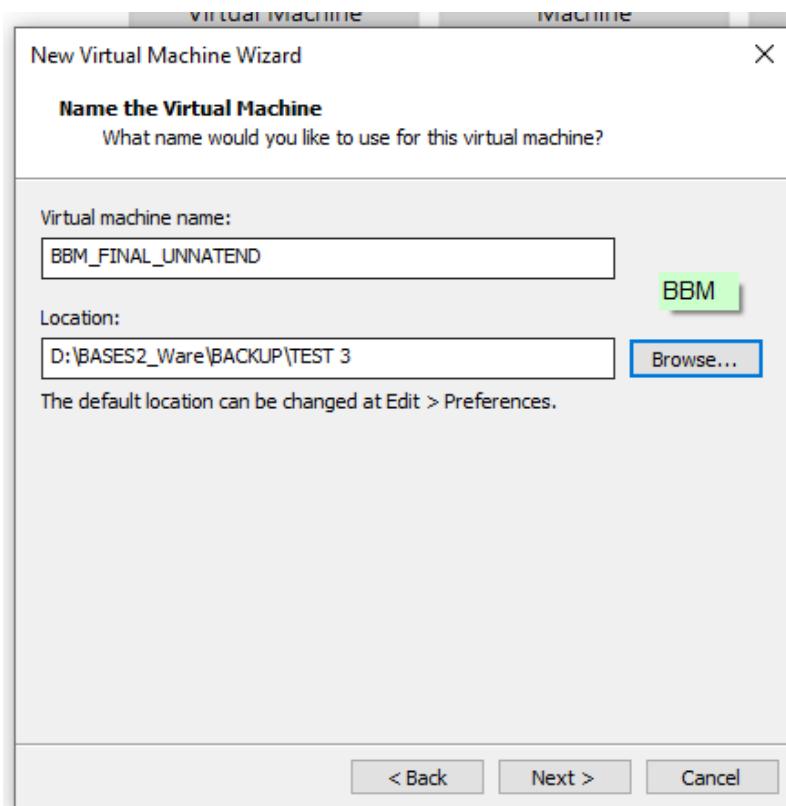
Dejo un registro del archivo que elegí:



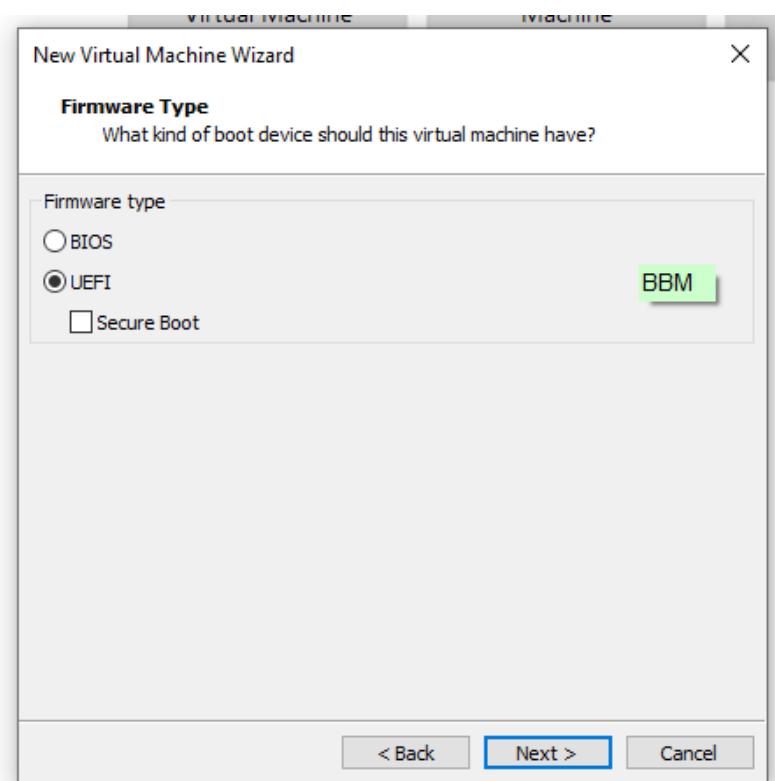
En la siguiente, elijo el sistema operativo (Microsoft Windows):



En la siguiente elijo el nombre y la localización de la máquina:



El tipo de *Firmware* será UEFI:



Las siguientes ventanas serán todas por defecto:

New Virtual Machine Wizard

Processor Configuration
Specify the number of processors for this virtual machine.

Processors
Number of processors: 2
Number of cores per processor: 1
Total processor cores: 2

New Virtual Machine Wizard

Memory for the Virtual Machine
How much memory would you like to use for this virtual machine?

Specify the amount of memory allocated to this virtual machine. The memory size must be a multiple of 4 MB.

64 GB	Memory for this virtual machine: 8192 MB
32 GB	
16 GB	
8 GB	
4 GB	
2 GB	
1 GB	
512 MB	
256 MB	
128 MB	
64 MB	
32 MB	
16 MB	
8 MB	
4 MB	

Maximum recommended memory: 27.8 GB
Recommended memory: 2 GB
Guest OS recommended minimum: 1 GB

New Virtual Machine Wizard

Network Type
What type of network do you want to add?

Network connection
 Use bridged networking
 Use network address translation (NAT)
 Use host-only networking
 Do not use a network connection

New Virtual Machine Wizard

Select I/O Controller Types
Which SCSI controller type would you like to use?

I/O controller types
SCSI Controller:
 BusLogic (Not available for 64-bit guests)
 LSI Logic (Not supported by Windows Server 2016)
 LSI Logic SAS (Recommended)

New Virtual Machine Wizard

Select a Disk Type
What kind of disk do you want to create?

Virtual disk type
 IDE
 SCSI (Recommended)
 SATA
 NVMe

New Virtual Machine Wizard

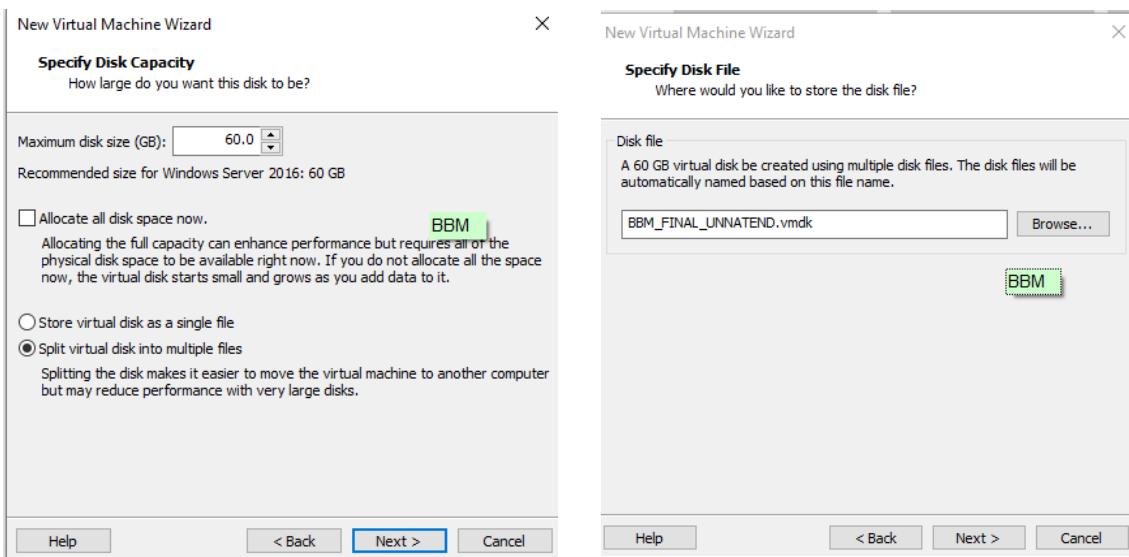
Select a Disk
Which disk do you want to use?

Disk
 Create a new virtual disk
A virtual disk is composed of one or more files on the host file system, which will appear as a single hard disk to the guest operating system. Virtual disks can easily be copied or moved on the same host or between BBM

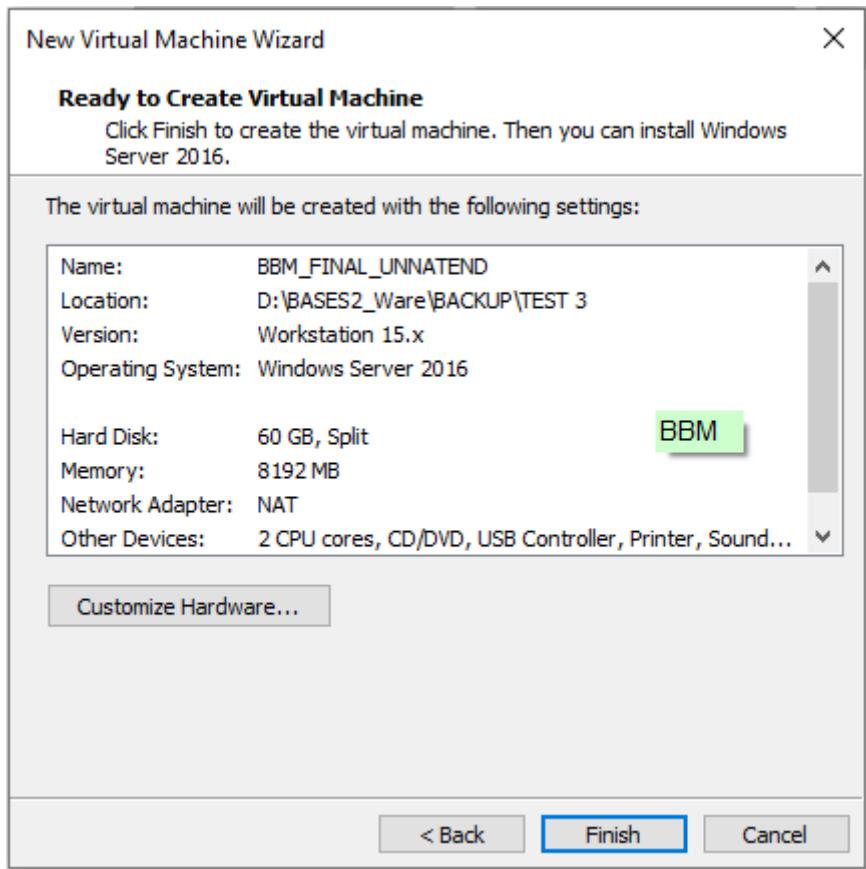
Use an existing virtual disk
Choose this option to reuse a previously configured disk.

Use a physical disk (for advanced users)
Choose this option to give the virtual machine direct access to a local hard disk. Requires administrator privileges.

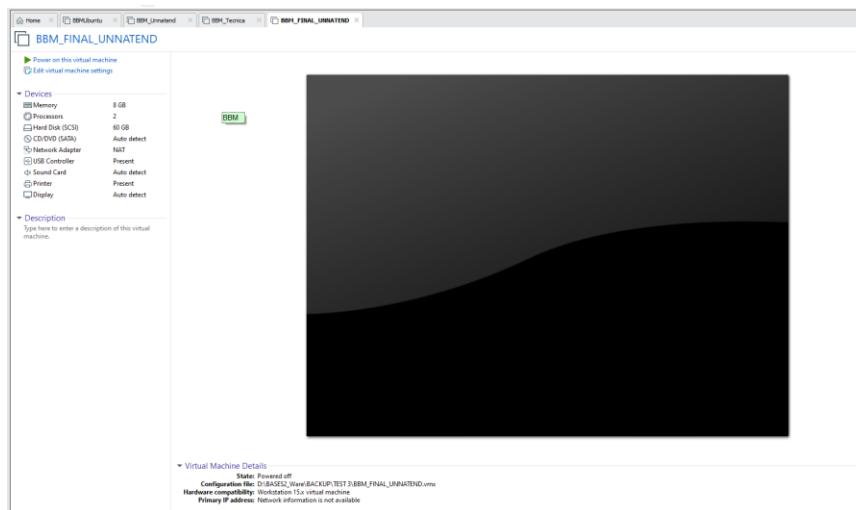
En la siguiente elijo el tamaño del disco. También por defecto junto al archivo de disco:



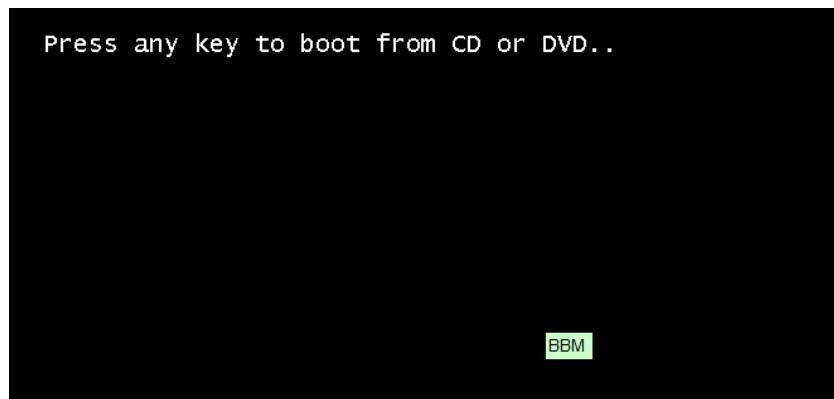
La última es un resumen. Le damos a Finish:



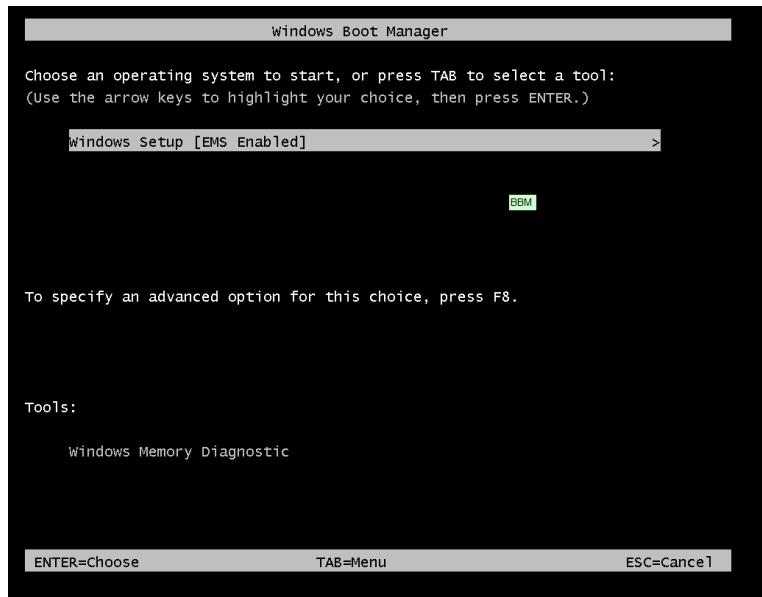
Ahora ponemos el CD en la máquina virtual y le damos a Power on this virtual machine:



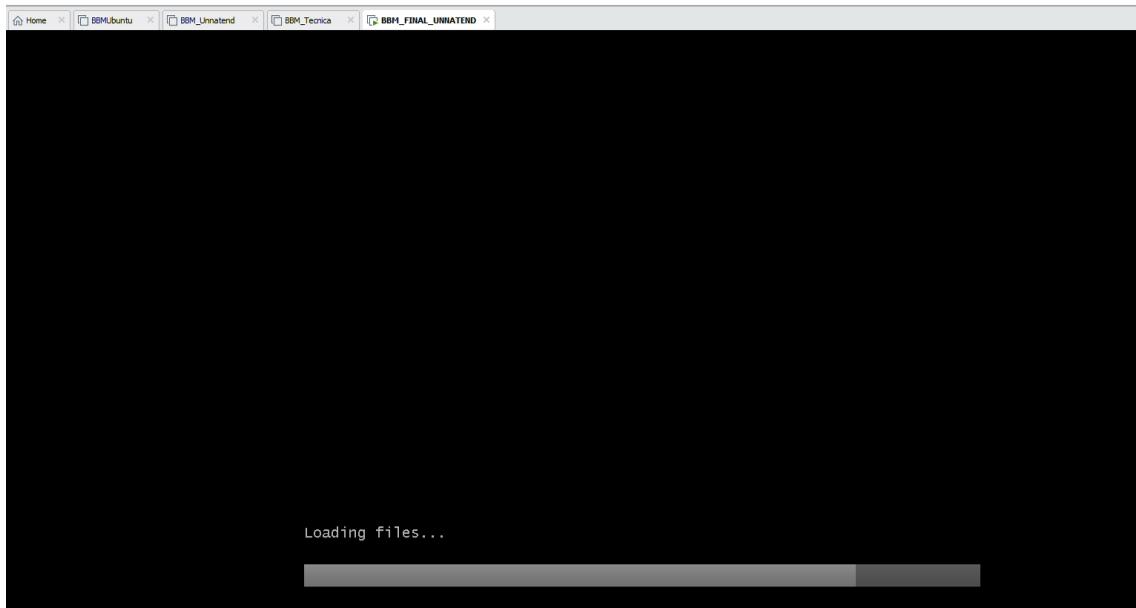
Vemos como ahora nos solicita la carga desde CD o DVD, pulsamos **Enter**.



Salta la ventana de advertencia de *Windows Setup*, le doy a **Enter**.



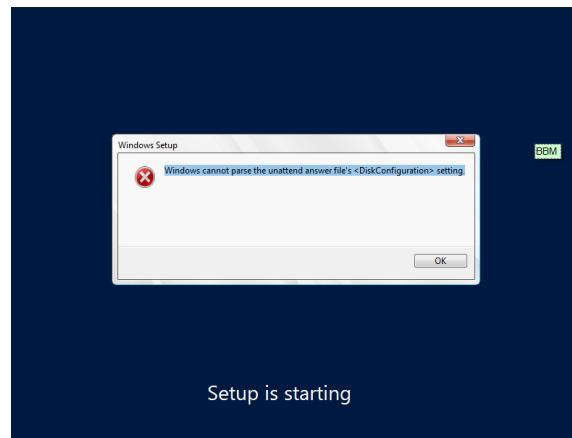
Vemos como empieza a cargar los archivos:



Vemos como carga normalmente:



Ahora me salta este error:



Reviso archivo autounatten.xml y veo que faltan apartado extend en archive configuration

```
<Disk wcm:action="add">
  <CreatePartitions>
    <CreatePartition wcm:action="add">
      <Extend>false</Extend>
      <Order>1</Order>
      <Size>300</Size>
      <Type>Primary</Type>
    </CreatePartition>
    <CreatePartition wcm:action="add">
      <Extend>false</Extend>
      <Order>2</Order>
      <Type>EFI</Type>
      <Size>200</Size>
    </CreatePartition>
    <CreatePartition wcm:action="add">
      <Extend>false</Extend>
      <Order>3</Order>
      <Size>128</Size>
      <Type>MSR</Type>
    </CreatePartition>
    <CreatePartition wcm:action="add">
      <Extend>false</Extend>
      <Order>4</Order>
      <Size>60000</Size>
      <Type>Primary</Type>
    </CreatePartition>
  </CreatePartitions>
```

Rearmo todo y en la instalación me sale este error:

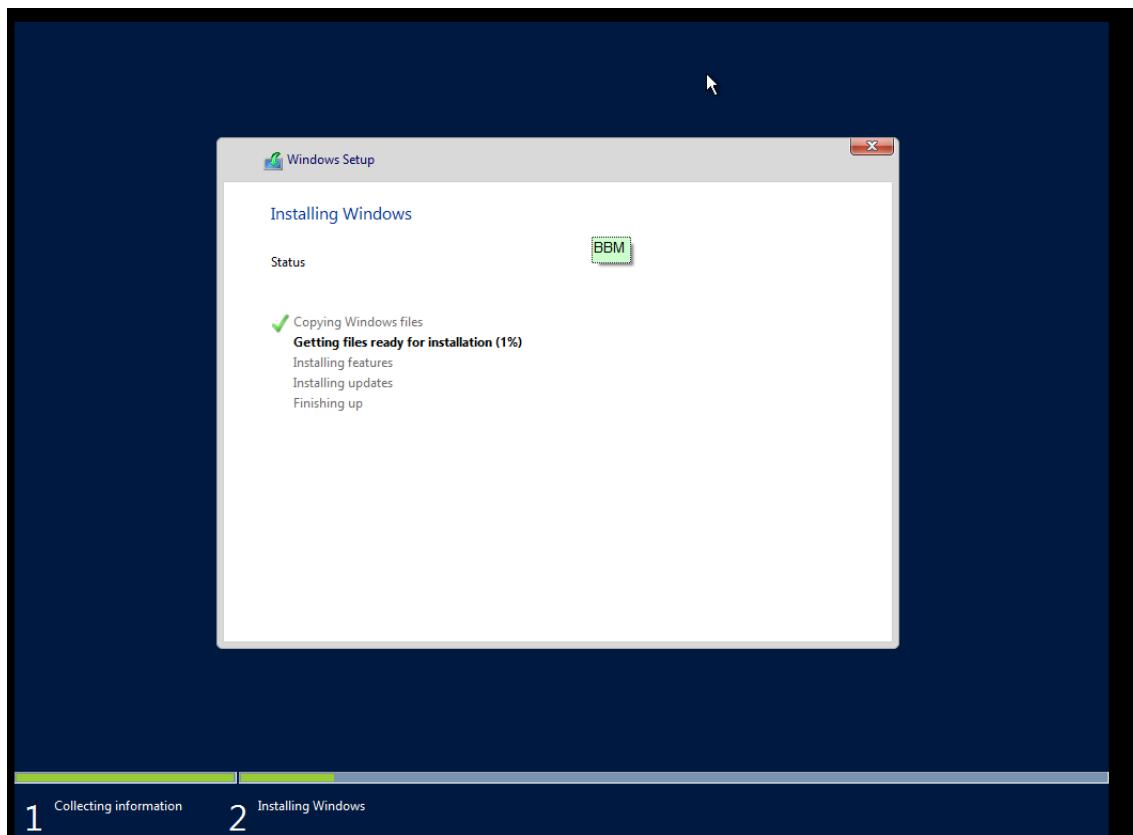


NOTA: Imagen sacada de internet

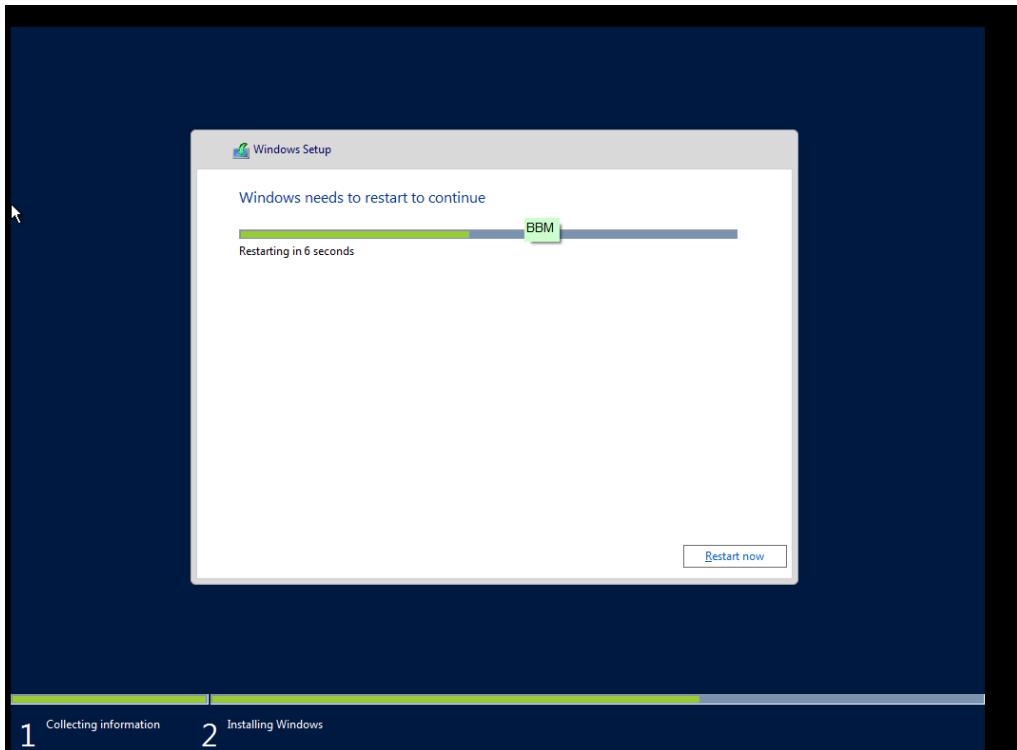
En el apartado de clave (Key) en Product Key, cambié la clave por la de un Windows Server Datacenter.

```
</component>
<component name="Microsoft-Windows-Setup" processorArchitecture="amd64" publicKeyToken="31bf3856ad364e35" xmlns:disk="http://schemas.microsoft.com/wmi/2005/disk" xmlns:processor="http://schemas.microsoft.com/wmi/2005/processor">
  <ImageInstall>
    <OSImage>
      <InstallTo>
        <DiskID>0</DiskID>
        <PartitionID>4</PartitionID>
      </InstallTo>
      <WillShowUI>Never</WillShowUI>
    </OSImage>
  </ImageInstall>
  <UserData>
    <AcceptEula>true</AcceptEula>
    <FullName></FullName>
    <Organization>BBM</Organization>
    <ProductKey>
      <WillShowUI>Never</WillShowUI>
      <Key>CB7KF-BWN84-R7R2Y-793K2-8XDDG</Key>
    </ProductKey>
  </UserData>
  <EnableFirewall>true</EnableFirewall>
  <DiskConfiguration>
    <Disk wcm:action="add">
```

Al iniciar ya lo carga todo normalmente:

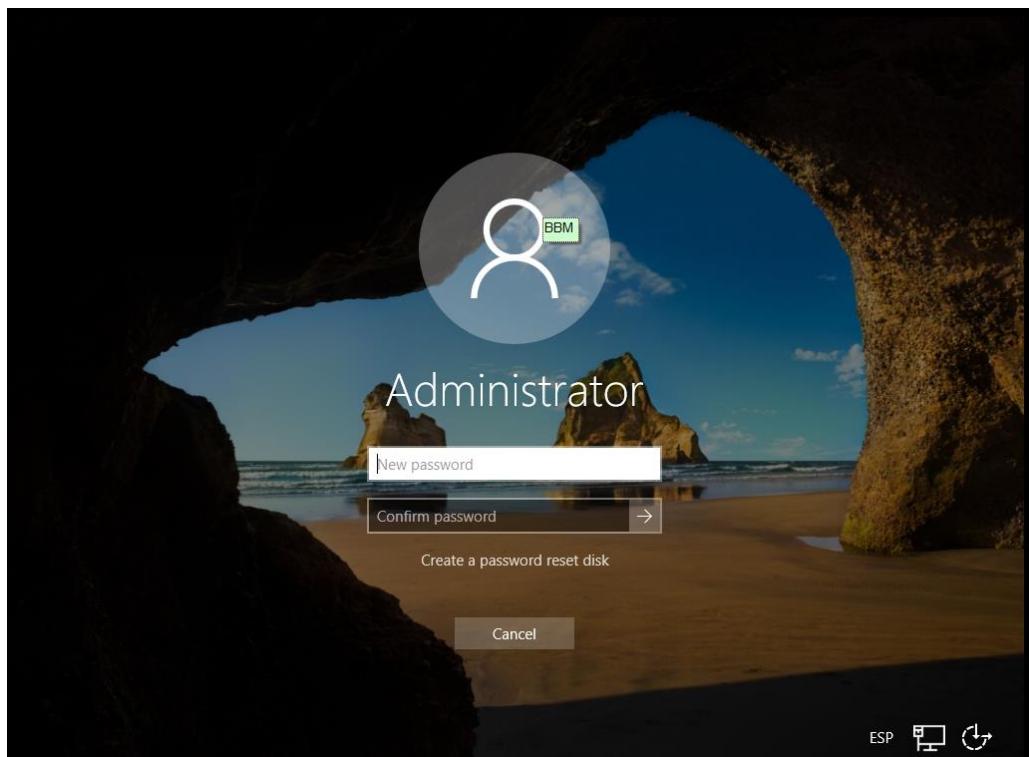


Se ve como hace las cargas y reinicios necesarios.

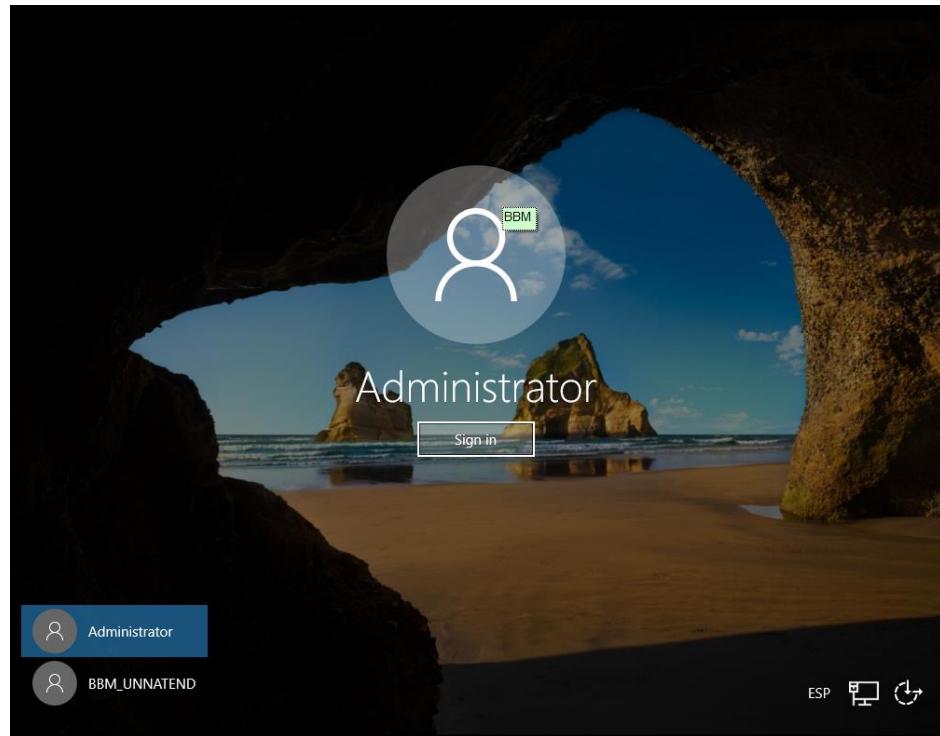


1 Collecting information 2 Installing Windows

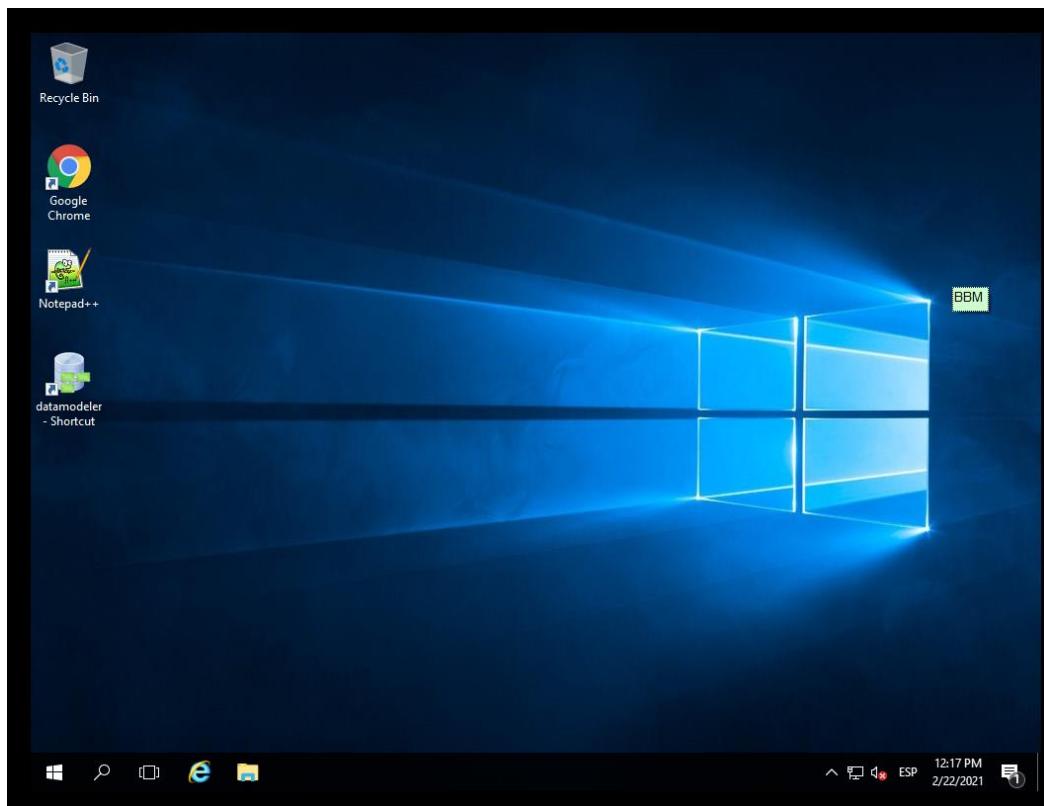
En la pantalla principal (aunque no debería) nos sale para cambiar la contraseña de Administrator.



Una vez la cambio, ya puedo entrar



Por último, pruebo a logarme con el user BBM_UNNATEND

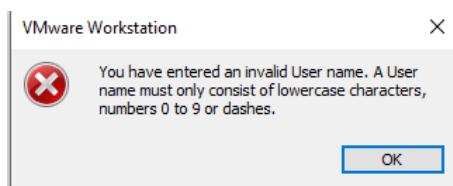
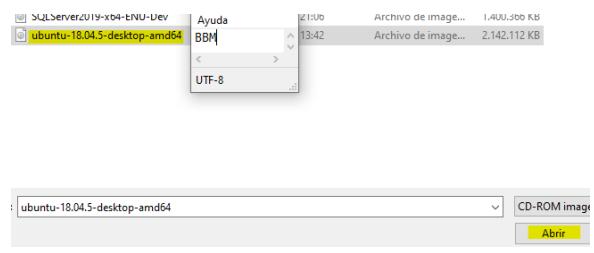


NOTA: Vemos en el escritorio que todo está tal cual lo instalamos.

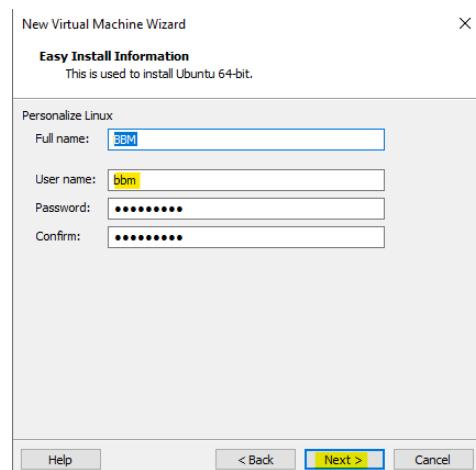
LINUX

Creación de MV e instalación de Ubuntu 18.04

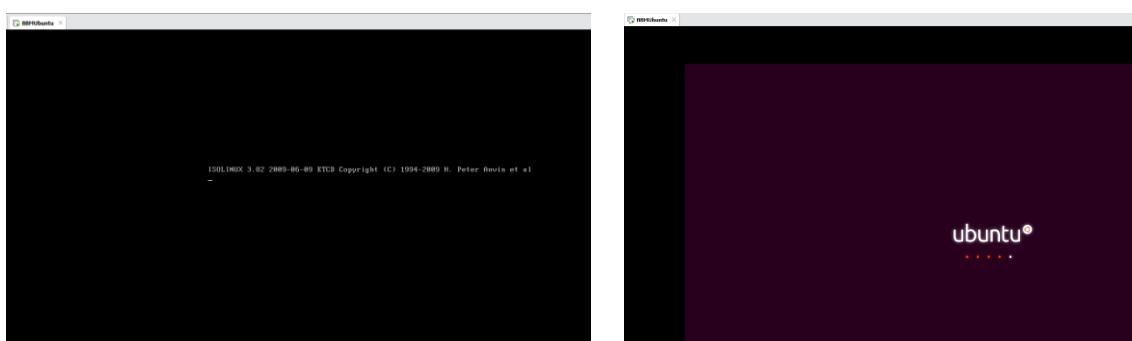
La creación de una máquina virtual ya está explicada en el [apartado 1.1.1](#) de este proyecto. Los pasos son exactamente los mismos en VMWARE, por lo tanto, la única diferencia remarcable en este aspecto es a la hora de elegir la ISO utilizada, que en este caso será **Ubuntu Desktop 18.04**.

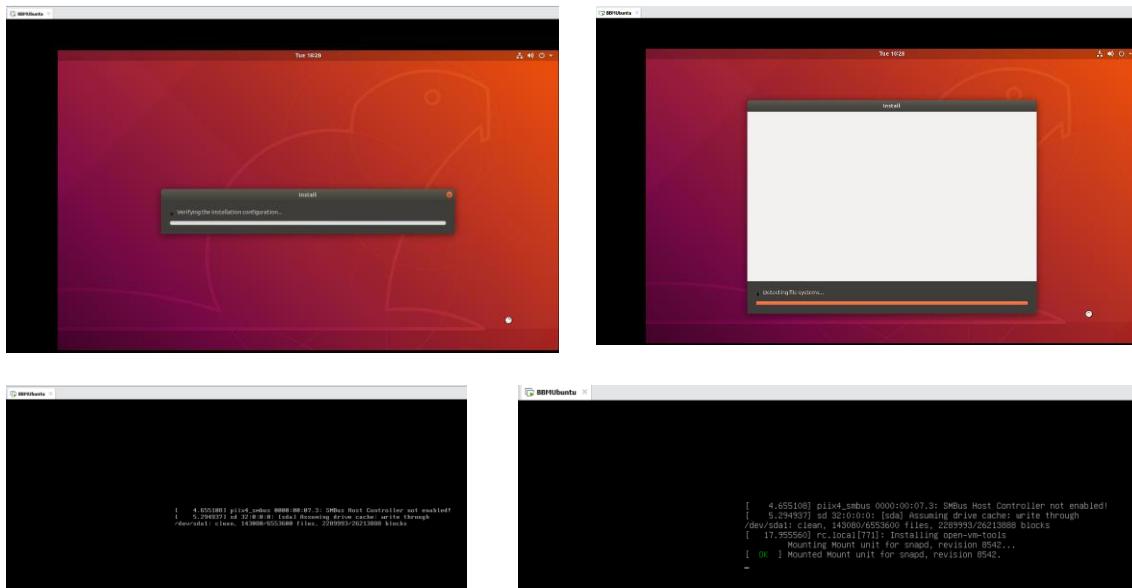


- ✗ La parte del nombre será como siempre las iniciales de los apellidos y el nombre (BBM). El usuario debido al error que dejo aquí, he tenido que ponerlo en minúsculas. La contraseña como siempre es la básica (*Abcd1234.*)

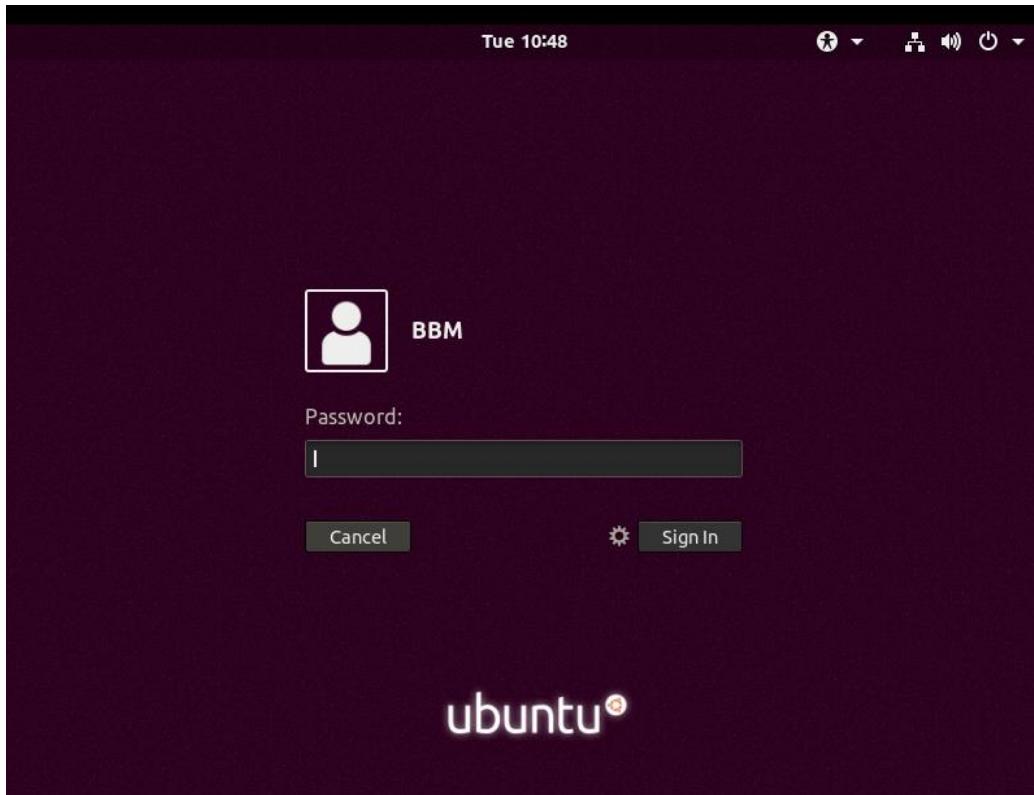


Las primeras pantallas de la instalación son de carga.





Como podemos ver, después de varias cargas, llegamos a la pantalla donde nos logaremos para empezar a trabajar



Configuración del firewall de Ubuntu

Al contrario que Windows, unir un FW de Ubuntu es muy sencillo y por línea de comandos. El comando principal es **sudo ufw enable** (habilita Ubuntu Firewall).

```
root@ubuntu:~# sudo ufw enable
Firewall is active and enabled on system startup
[...]
```

Para agregar las reglas tenemos dos comandos: **allow** para **añadir** reglas y **delete** para borrarlas precedido de **sudo ufw**. Sabiendo esto, añadir puertos es a esa línea de comandos en **número de puerto/protocolo**. Así tenemos:

- **sudo ufw allow 1434/tcp**

```
root@ubuntu:~# sudo ufw allow 1434/tcp
Rule added
Rule added (v6)
```

- **sudo ufw allow 1434/udp**

```
root@ubuntu:~# sudo ufw allow 1434/udp
Rule added
Rule added (v6)
```

- **sudo ufw allow 1433/tcp**

```
root@ubuntu:~# sudo ufw allow 1433/tcp
Rule added
Rule added (v6)
```

- **sudo ufw allow 1433/udp**

```
root@ubuntu:~# sudo ufw allow 1433/udp
Rule added
Rule added (v6)
```

NOTA: Como se ve en las imágenes, tanto lo hace en **IPv4** como en **IPv6**.

Por último, agregamos la regla que permite la **subred**. En este caso el comando es **sudo ufw in on <número de puerto> to <red>**:

sudo ufw in on ens36 to 192.168.0.0/29

```
root@ubuntu:~# sudo ufw allow in on ens36 to 192.168.0.0/29
Rule added
```

NOTA: Para saber el numero de puerto, hacemos **ip a** y por la **MAC de la interfaz** podemos distinguir cual es

```
root@ubuntu:/home/administrador# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:0c:29:1d:24:d2 brd ff:ff:ff:ff:ff:ff
    inet 192.168.139.133/24 brd 192.168.139.255 scope global ens33
        valid_lft forever preferred_lft forever
    inet6 fe80::20c:29ff:fe1d:24d2/64 scope link
        valid_lft forever preferred_lft forever
3: ens36: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:0c:29:1d:24:dc brd ff:ff:ff:ff:ff:ff
    inet 192.168.0.4/29 brd 192.168.0.7 scope global ens36
        valid_lft forever preferred_lft forever
    inet6 fe80::20c:29ff:fe1d:24dc/64 scope link
        valid_lft forever preferred_lft forever
root@ubuntu:/home/administrador#
```

Configuración de la red de Ubuntu

Entramos en el terminal y hacemos el comando ip a.

```
administrador@ubuntu:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:0c:29:1d:24:d2 brd ff:ff:ff:ff:ff:ff
    inet 192.168.139.133/24 brd 192.168.139.255 scope global ens33
        valid_lft forever preferred_lft forever
    inet6 fe80::20c:29ff:fe1d:24d2/64 scope link
        valid_lft forever preferred_lft forever
3: ens36: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:0c:29:1d:24:dc brd ff:ff:ff:ff:ff:ff
    inet 192.168.0.4/29 brd 192.168.0.7 scope global ens36
        valid_lft forever preferred_lft forever
    inet6 fe80::20c:29ff:fe1d:24dc/64 scope link
        valid_lft forever preferred_lft forever
```

Dentro del propio terminal, inicio sesión en el Ubuntu 18.04 con el usuario root. Para ello, introducimos el comando sudo su. Nos va a pedir la contraseña, metemos la de nuestro PC (en este caso Abc1234.)

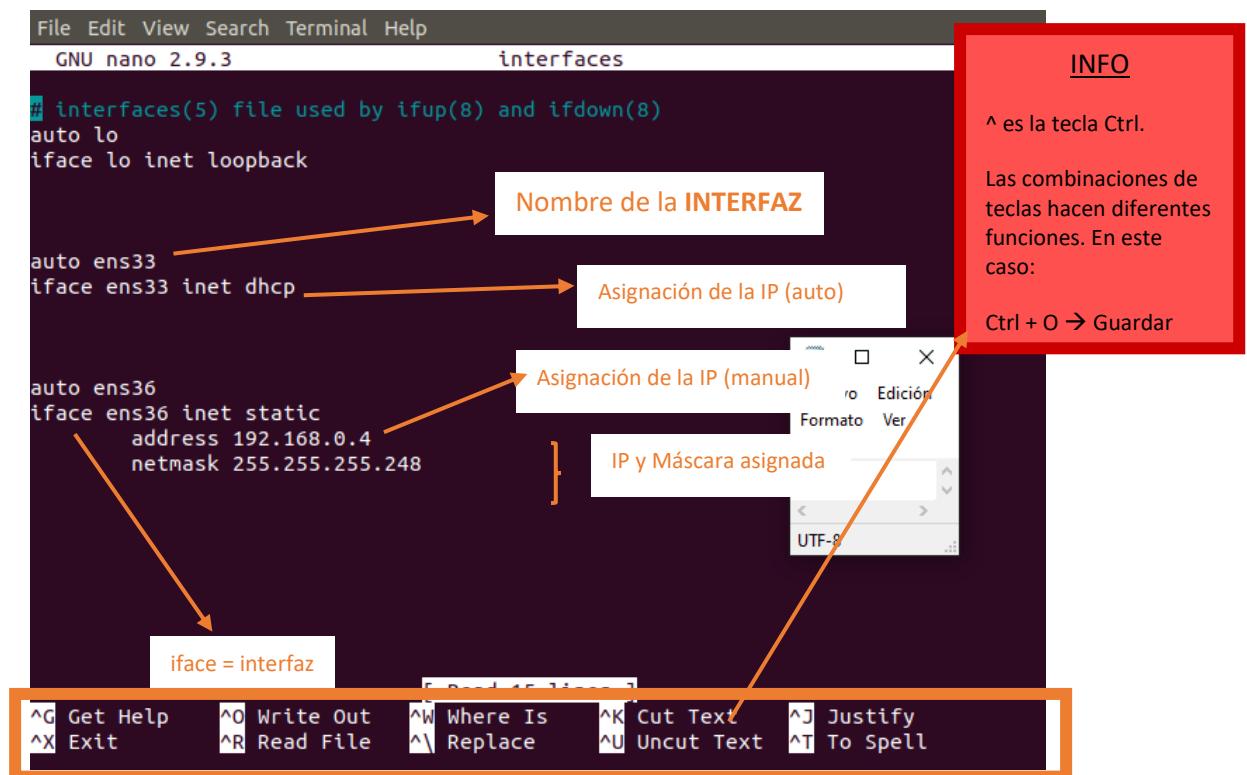
```
administrator@ubuntu:~$ sudo su
[sudo] password for administrator:
root@ubuntu:/home/administrator#
```

Nos pasamos al directorio network con el comando `cd /etc/network`. En él, cogemos el archivo *interfaces* (que es un archivo de texto) y le hacemos una copia por si necesitamos hacer un backup. Lo hacemos mediante el comando `cp <source> <destination>` (en este caso `cp interfaces interfaces.old`).

```
root@ubuntu:~# cd /etc/network
root@ubuntu:/etc/network# cp interfaces interfaces.old
```

Una vez hecha la copia, con el comando `nano interfaces` editamos el archivo.

```
root@debian:/etc/network# nano interfaces
```



Guardamos nuestro fichero (**Ctrl + O**) y lo guardamos como el nombre *interfaces*. Acto seguido salimos (**Ctrl + X**).



```
File Name to Write: interfaces
^G Get Help      M-D DOS Format    M-A Append     M-B Backup File
^C Cancel        M-M Mac Format    M-P Prepend   ^T To Files
^W Where Is      ^R Read File     ^K Cut Text    ^J Justify
^V Replace       ^U Uncut Text   ^X Exit        ^Y To Spell
```

```
root@ubuntu:/etc/network# more interfaces
# interfaces(5) file used by ifup(8) and ifdown(8)
auto lo
iface lo inet loopback

auto ens33
iface ens33 inet dhcp

auto ens36
iface ens36 inet static
    address 192.168.0.4
    netmask 255.255.255.248
```

Vemos la configuración del archivo con `more interfaces`:

Reiniciamos el servicio para que se apliquen todos los cambios hechos hasta ahora. Se hace con el comando sudo /etc/init.d/network-manager restart.

```
root@ubuntu:~# sudo /etc/init.d/network-manager restart
[ ok ] Restarting network-manager (via systemctl): network-manager.service. BBM
```

Por último, hacemos un ping a la máquina del servidor y al cliente de Windows 10 con el comando ping <IP> -c 2 (para que sólo lance dos pines):

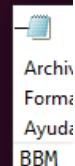
```
root@ubuntu:/etc# ping 192.168.0.2 -c 2
PING 192.168.0.2 (192.168.0.2) 56(84) bytes of data.
64 bytes from 192.168.0.2: icmp_seq=1 ttl=128 time=0.335 ms
64 bytes from 192.168.0.2: icmp_seq=2 ttl=128 time=0.982 ms

--- 192.168.0.2 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1007ms
rtt min/avg/max/mdev = 0.335/0.658/0.982/0.324 ms
root@ubuntu:/etc# ping 192.168.0.3 -c 2
PING 192.168.0.3 (192.168.0.3) 56(84) bytes of data.
64 bytes from 192.168.0.3: icmp_seq=1 ttl=128 time=0.564 ms
64 bytes from 192.168.0.3: icmp_seq=2 ttl=128 time=1.20 ms

--- 192.168.0.3 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1003ms
rtt min/avg/max/mdev = 0.564/0.885/1.207/0.322 ms
```

INFO

Otras opciones del comando network-manager son *start* y *stop*. A efectos prácticos es lo mismo que *restart*.



Archiv
Format
Ayuda
BBM

Unir Ubuntu al dominio

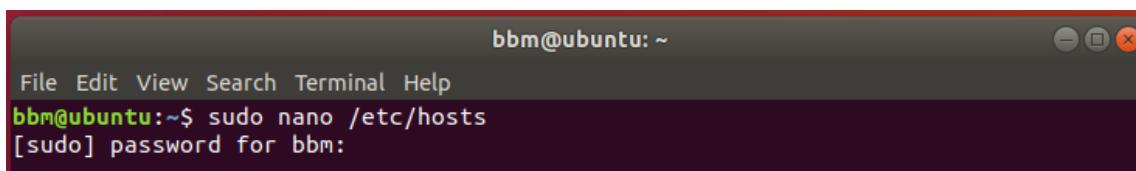
La tarea es un poco larga. En una primera parte, se resolverán los siguientes aspectos:

- Configurar las características del equipo.
- Instalar el software necesario.
- Configurar *Kerberos*.
- Comprobar el funcionamiento de *Kerberos*.
- Unir el cliente al dominio.

Y una segunda que se encargará de lo siguiente:

- Configurar *SSSD*.
- Ajustar el comportamiento de *PAM* para iniciar sesión con usuarios del dominio.
- Convertir al administrador del dominio en administrador local.
- Reiniciar el equipo y comprobar el inicio de sesión gráfico.
- Iniciar sesión con la cuenta del dominio.

Para configurar las características del equipo, se comienza editando en archivo **/etc/hosts** con el editor de textos **nano**: `sudo nano /etc/hosts` (la primera vez tenemos que introducir la contraseña **Abcd1234**.). Como es lógico, para este tipo de operaciones debemos utilizar privilegios administrativos.



A screenshot of a terminal window titled "bbm@ubuntu: ~". The window has a dark theme with white text. The title bar shows the user "bbm@ubuntu" and the prompt "~". Below the title bar is a menu bar with "File", "Edit", "View", "Search", "Terminal", and "Help". The main area of the terminal shows the command `bbm@ubuntu:~$ sudo nano /etc/hosts` followed by a password prompt "[sudo] password for bbm:". The background of the terminal window is dark, and the window has a standard Linux-style title bar with minimize, maximize, and close buttons.

Lo siguiente será añadir una nueva línea en el archivo que relacione la dirección IP del cliente con su nombre de equipo, incluyendo también el nombre completo dentro del dominio: **192.168.0.2 server-2016-a server-2016-a.bbminspace.local**. Cuando estemos listos, pulsamos la combinación de teclas **Ctrl** + **X** para cerrar el editor de textos.

```
bbm@ubuntu: ~
File Edit View Search Terminal Help
GNU nano 2.9.3          /etc/hosts          Modified
127.0.0.1      localhost
127.0.1.1      ubuntu
192.168.0.2    server-2016-a server-2016-a.bbmaspace.local

# The following lines are desirable for IPv6 capable hosts
::1      ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters

Save modified buffer? (Answering "No" will DISCARD changes.)
Y Yes
N No
^C Cancel
```

Pulsamos la tecla **Intro** para conservar el mismo nombre del archivo:

```
bbm@ubuntu: ~
File Edit View Search Terminal Help
GNU nano 2.9.3          /etc/hosts          Modified
127.0.0.1      localhost
127.0.1.1      ubuntu
192.168.0.2    server-2016-a server-2016-a.bbmaspace.local

# The following lines are desirable for IPv6 capable hosts
::1      ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters

Help
File Name to Write: /etc/hosts
^G Get Help      M-D DOS Format      M-A Append      M-B Backup File
^C Cancel        M-M Mac Format      M-P Prepend     ^T To Files
```

Finalmente, para comprobar que los cambios han tenido éxito, probamos a ejecutar la orden *ping* con el nombre del servidor:

```
root@ubuntu:~# ping 192.168.0.2 -c 2
PING 192.168.0.2 (192.168.0.2) 56(84) bytes of data.
64 bytes from 192.168.0.2: icmp_seq=1 ttl=128 time=1.07 ms
64 bytes from 192.168.0.2: icmp_seq=2 ttl=128 time=0.708 ms
--- 192.168.0.2 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1002ms
rtt min/avg/max/mdev = 0.708/0.889/1.070/0.181 ms
root@ubuntu:~#
```

Una vez que estamos seguros de que la comunicación entre el cliente y el servidor funciona correctamente, el siguiente paso consistirá en instalar el software necesario para realizar la tarea.

Para lograrlo, comenzaremos por actualizar el sistema operativo con las últimas versiones de los paquetes disponibles en los repositorios. Algo tan sencillo como utilizar la siguiente orden: `sudo apt update && sudo apt upgrade`.

```
root@ubuntu:~# sudo apt update && sudo apt upgrade
Hit:1 http://security.ubuntu.com/ubuntu bionic-security InRelease
Hit:2 http://packages.microsoft.com/repos/vscode stable InRelease
Hit:3 http://us.archive.ubuntu.com/ubuntu bionic InRelease
Get:4 http://us.archive.ubuntu.com/ubuntu bionic-updates InRelease [88.7 kB]
Hit:5 https://packages.microsoft.com/ubuntu/18.04/mssql-server-2019 bionic InRe
lease
Get:6 http://us.archive.ubuntu.com/ubuntu bionic-backports InRelease [74.6 kB]
Fetched 163 kB in 2s (75.3 kB/s)
Reading package lists... Done
```

Al hacerlo, recibiremos un detalle con todos los paquetes que van a actualizarse y un resumen final con el número de paquetes que se actualizarán, los paquetes nuevos que se instalarán y los que se eliminarán de la instalación. También se incluye la cantidad de información que se descargará de los repositorios y la capacidad final que se utilizará en nuestro disco duro. Cuando salga el mensaje, escribimos `Y` y pulsamos Intro.

```
Get:6 http://us.archive.ubuntu.com/ubuntu bionic-backports InRelease [74.6 kB]
Fetched 163 kB in 2s (75.3 kB/s)
Reading package lists... Done
Building dependency tree
Reading state information... Done
50 packages can be upgraded. Run 'apt list --upgradable' to see them.
Reading package lists... Done
Building dependency tree
Reading state information... Done
Calculating upgrade... Done
The following packages will be upgraded:
  apport apport-gtk base-files gdb gdbserver grub-common grub-pc grub-pc-bin
  grub2-common initramfs-tools initramfs-tools-bin initramfs-tools-core
  libc-bin libc6 libc6-dbg libcryptsetup12 libdevicemapper libinput libinput10
  libnss-myhostname libnss-systemd libpam-modules libpam-modules-bin
  libpam-runtime libpam-systemd libpam0g libpcap0.8 libsane-common libsane1
  libsystemd0 libudev1 libwhoopsie0 locales multiarch-support python3-apport
  python3-distupgrade python3-problem-report sane-utils snapd sudo systemd
  systemd-sysv ubuntu-desktop ubuntu-minimal ubuntu-release-upgrader-core
  ubuntu-release-upgrader-gtk ubuntu-standard udev whoopsie wpasupplicant
50 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
Need to get 13.6 MB/49.6 MB of archives.
After this operation, 5,275 kB of additional disk space will be used.
Do you want to continue? [Y/n]
```

Cuando acabe la actualización, se pasa a instalar los siguientes paquetes:

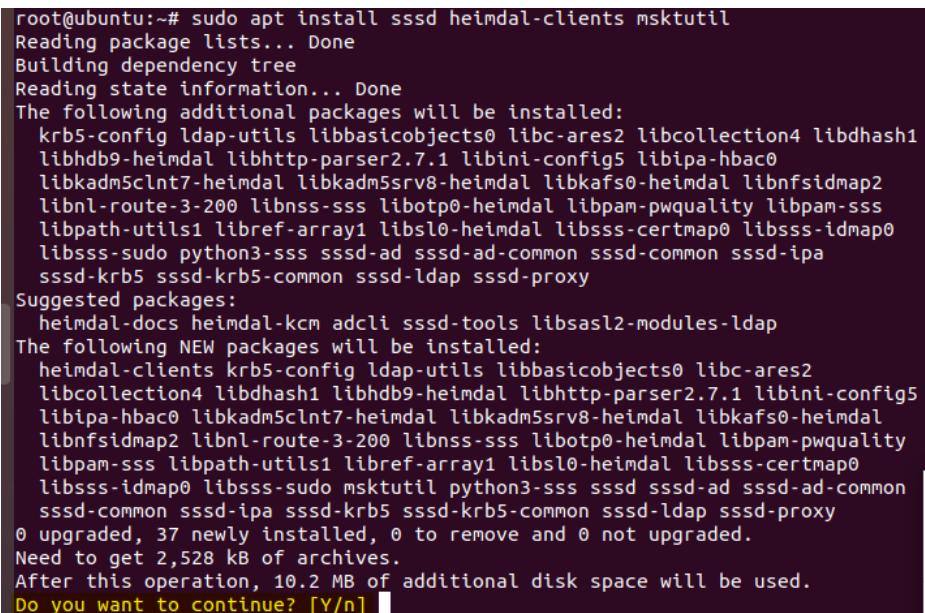
- ***sssd (System Security Services Daemon)***: Administra los mecanismos de autenticación y el acceso a directorios remotos.
- ***heimdal-clients***: Se trata de una implementación libre de Kerberos 5 creada con la intención de ser compatible con el protocolo Kerberos.
- ***msktutil***: La utilidad que obtiene y administra los keytabs (archivo que contiene parejas de entidades de seguridad y claves cifradas de Kerberos) de Kerberos en un entorno de Microsoft Active Directory.

Para instalar todos los paquetes en una sola operación, sólo tenemos que utilizar la siguiente orden: sudo apt install sssd heimdal-clients msktutil.



```
root@ubuntu:~# sudo apt install sssd heimdal-clients msktutil
```

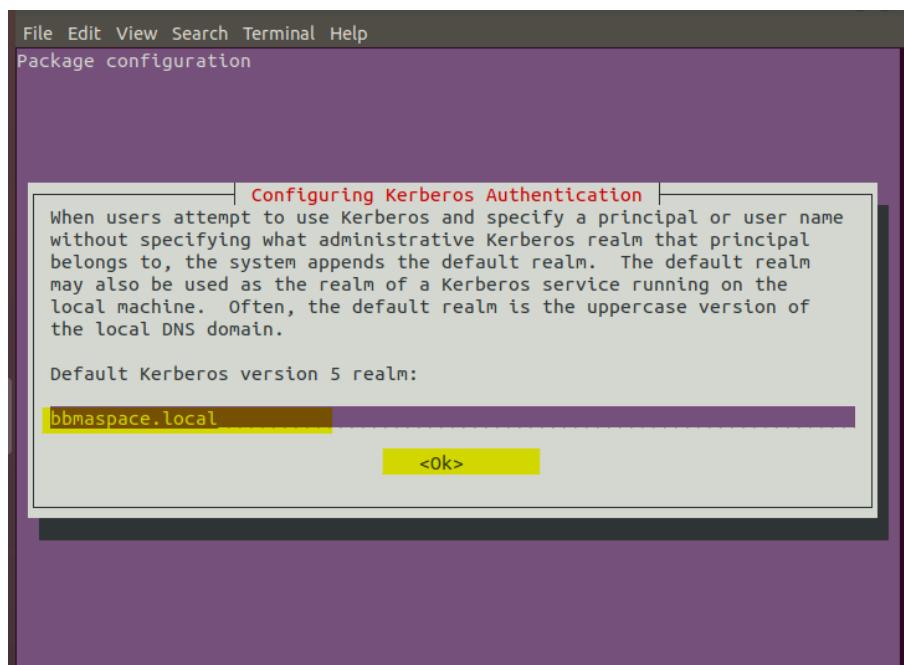
Nos aparecerá nuevamente un listado con paquetes de actualizaciones. Lo aceptamos.



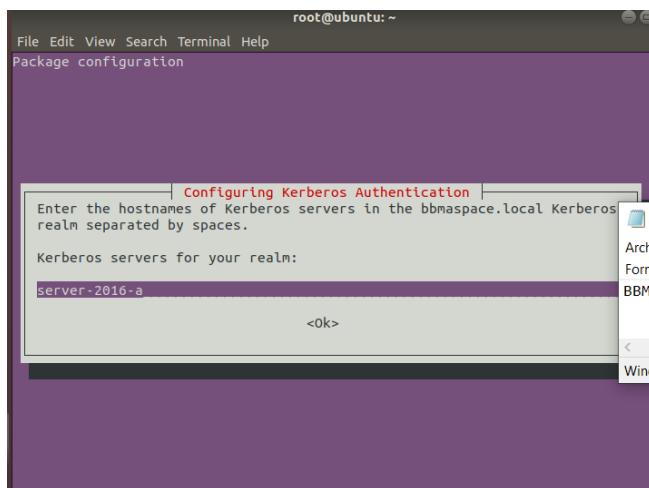
```
root@ubuntu:~# sudo apt install sssd heimdal-clients msktutil
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  krb5-config ldap-utils libbasicobjects0 libc-ares2 libcollection4 libdhash1
  libhdb9-heimdal libhttp-parser2.7.1 libini-config5 libipa-hbac0
  libkadm5clnt7-heimdal libkadm5srv8-heimdal libkafs0-heimdal libnfsidmap2
  libnl-route-3-200 libnss-sss libotp0-heimdal libpam-pwquality libpam-sss
  libpath-utils1 libref-array1 libsl0-heimdal libsss-certmap0 libsss-idmap0
  libsss-sudo python3-sss sssd-ad sssd-ad-common sssd-common sssd-ipa
  sssd-krb5 sssd-krb5-common sssd-ldap sssd-proxy
Suggested packages:
  heimdal-docs heimdal-kcm adcli sssd-tools libsasl2-modules-ldap
The following NEW packages will be installed:
  heimdal-clients krb5-config ldap-utils libbasicobjects0 libc-ares2
  libcollection4 libdhash1 libhdb9-heimdal libhttp-parser2.7.1 libini-config5
  libipa-hbac0 libkadm5clnt7-heimdal libkadm5srv8-heimdal libkafs0-heimdal
  libnfsidmap2 libnl-route-3-200 libnss-sss libotp0-heimdal libpam-pwquality
  libpam-sss libpath-utils1 libref-array1 libsl0-heimdal libsss-certmap0
  libsss-idmap0 libsss-sudo msktutil python3-sss sssd sssd-ad sssd-ad-common
  sssd-common sssd-ipa sssd-krb5 sssd-krb5-common sssd-ldap sssd-proxy
0 upgraded, 37 newly installed, 0 to remove and 0 not upgraded.
Need to get 2,528 kB of archives.
After this operation, 10.2 MB of additional disk space will be used.
Do you want to continue? [Y/n]
```



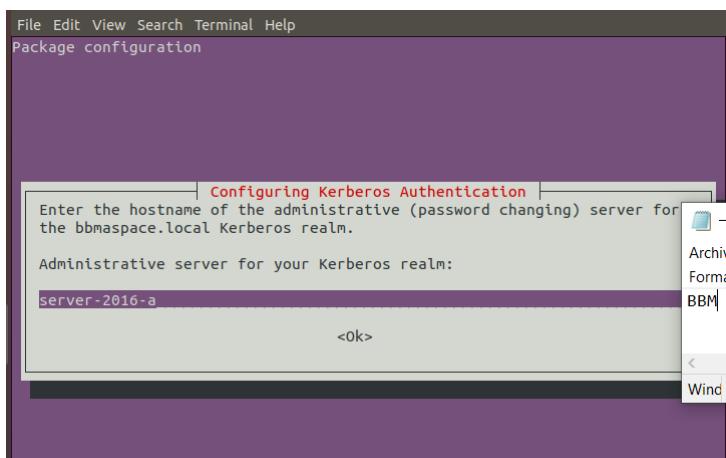
Cuando llegue el momento de instalar Kerberos, pregunta por el nombre del dominio al que vamos a unirnos: Escribo bbmaspace.local.



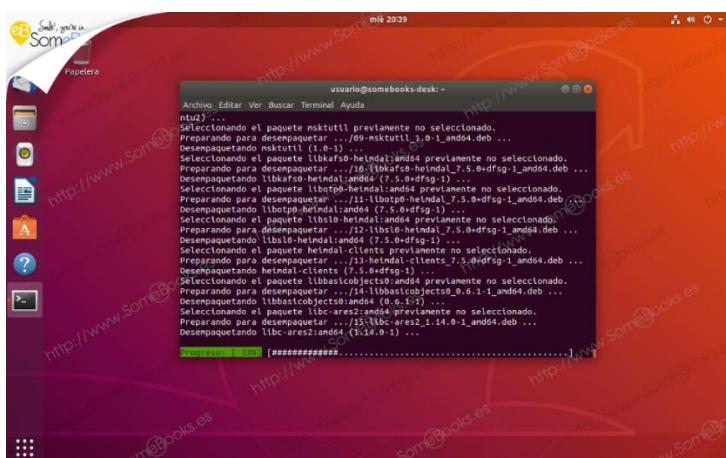
Después, el instalador solicita el nombre de los servidores de Kerberos para nuestro **realm**. En este caso, se refiere al controlador de dominio, (**server-2016-a**)



Por último, solicita el nombre del equipo que actúa como servidor administrativo para la autenticación en el dominio. **Aceptamos**.



Después de esto, la instalación continuará un poco más, pero sin necesitar que aportemos más información. Solo tenemos que esperar hasta que se complete.



Ahora para configurar Kerberos, hay que añadir más información a la configuración para que se comporte de forma adecuada en el dominio. Se comienza por cambiar

el nombre del archivo de configuración, para poder volver a los parámetros originales si fuese preciso: `sudo mv /etc/krb5.conf /etc/krb5.conf.default`.

```
root@ubuntu:~# sudo mv /etc/krb5.conf /etc/krb5.conf.default
```

BBM

Con el editor `nano` se crea un nuevo archivo de configuración: `sudo nano /etc/krb5.conf`.

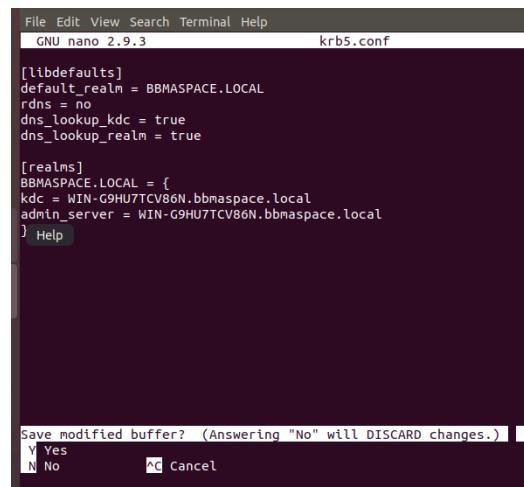
```
root@ubuntu:~# sudo nano /etc/krb5.conf
```

BBM

Al hacerlo, abrimos un documento vacío, donde se escribe lo siguiente:

```
[libdefaults]
default_realm = BBMSPACE.LOCAL
rdns = no
dns_lookup_kdc = true
dns_lookup_realm = true

[realms]
BBMSPACE.LOCAL = {
kdc = WIN-G9HU7TCV86N.bbmaspace.local
admin_server = WIN-G9HU7TCV86N.bbmaspace.local
}
```



NOTA: Cuando estemos listos, se pulsa la combinación de teclas `Ctrl + X` para cerrar el editor de textos.

Para comprobar que la autenticación en el dominio, se produce de forma satisfactoria, se utiliza el comando `kinit` y la cuenta *Administrator* del propio dominio: `kinit administrador`.

```
root@ubuntu:~# sudo nano /etc/krb5.conf
root@ubuntu:~# kinit administrador
administrador@BBMSPACE.LOCAL's Password: 
```

Formato

BBM

Si todo va bien, Kerberos pide la contraseña de la cuenta.

```
root@ubuntu:~# sudo nano /etc/krb5.conf
root@ubuntu:~# kinit administrador
administrador@BBMSPACE.LOCAL's Password: 
```

Formato

BBM

Si la salida no nos ofrece ningún tipo de error, es porque el proceso de autenticación ha funcionado correctamente. Utilizando el comando `klist` para estar completamente seguros.

klist muestra la información sobre el *ticket de autenticación*, incluida la fecha y hora de concesión, de caducidad, la cuenta y el dominio que estamos empleando.

```
root@ubuntu:/etc# klist
Credentials cache: FILE:/tmp/krb5cc_0
    Principal: administrator@BBMASPACE.LOCAL

    Issued          Expires          Principal
Jan  3 03:46:13 2021 Jan  3 13:46:13 2021 krbtgt/bbminspace.local@BBMASPACE
CAL
root@ubuntu:/etc#
```

Una vez que está asegurado el funcionamiento de Kerberos, la comprobación definitiva será unir el equipo al dominio de *Windows Server 2016*. Para lograrlo, se usa el comando **msktutil** con la siguiente sintaxis: **msktutil -N -c -b 'CN=Computers' -s BBMASPACE/bbminspace.local -k bbmkeytab.keytab --computer-name BBMUBUNTU --upn BBMUBUNTUSER --server WIN-G9HU7TCV86N.bbminspace.local --user-creds-only**.

```
root@ubuntu:/etc# msktutil -N -c -b 'CN=COMPUTERS' -s BBMASPACE/bbminspace.local
root@ubuntu:/etc# msktutil -N -c -b 'CN=COMPUTERS' -s BBMASPACE/bbminspace.local
    -k my-keytab.keytab --computer-name BBMASPACE --upn BBMASPACE$ --server bbmasp
ace.local --user-creds-only
```

Para entender el comando, dejo un resumen de los argumentos empleados:

- **-N** le dice al comando que no realiza búsquedas DNS inversas para canonizar el nombre del dominio.
- **-c** indica que debe crearse un *keytab* predeterminado.
- **-b** establece el contenedor donde debe crearse la cuenta para el equipo dentro del dominio.
- **-s** incluye el equipo responsable de servicio para añadir a la cuenta. tiene la forma <servicio>/<nombre de host>.
- **-k** contiene el nombre del archivo que se usará para almacenar el *keytab*.
- **-computer-name** indica el nombre con el que se creará la cuenta del equipo en el directorio.
- **-upn** establece el nombre de usuario principal del equipo. En su lugar usaremos el nombre del equipo, lo que se especifica incluyéndolo seguido de un carácter \$.
- **-server** identifica al controlador de dominio
- **--user-creds-only** fuerza a que se utilicen las credenciales de usuario.

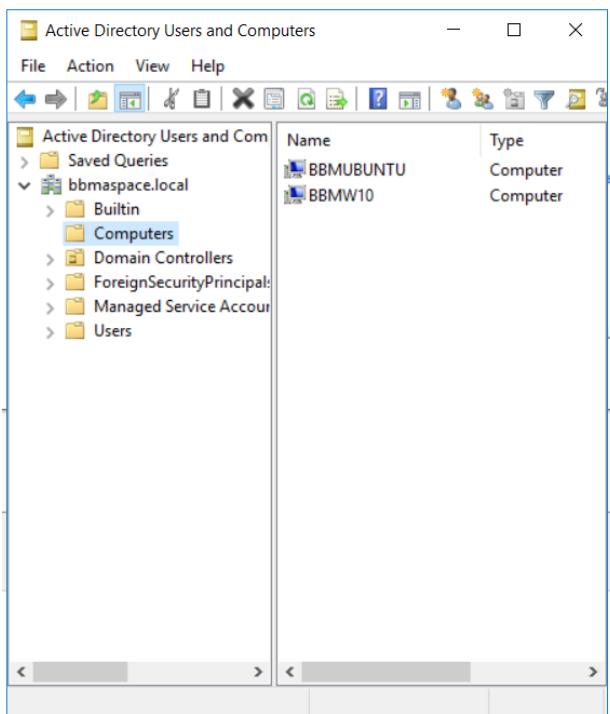
Como en el argumento **-s** se incluye el *nombre DNS* del equipo, a continuación, se repite la misma orden usando en *nombre NetBIOS*: **msktutil -N -c -b 'CN=Computers' -s BBMASPACE/bbminspace.local -k bbmkeytab.keytab --computer-name BBMUBUNTU --upn BBMUBUNTUSER --server WIN-G9HU7TCV86N.bbminspace.local --user-creds-only**.

```
root@ubuntu:/etc# msktutil -N -c -b 'CN=Computers' -s BBMASPACE/bbminspace.local
-k bbmkeytab.keytab --computer-name BBMUBUNTU --upn BBMUBUNTUSER --server WIN-
G9HU7TCV86N.bbminspace.local --user-creds-only
root@ubuntu:/etc#
root@ubuntu:/etc# msktutil -N -c -b 'CN=Computers' -s BBMASPACE/bbminspace.local
-k bbmkeytab.keytab --computer-name BBMUBUNTU --upn BBMUBUNTUSER --server WIN-
G9HU7TCV86N.bbminspace.local --user-creds-only
root@ubuntu:/etc#
```

Para completar la tarea, se eliminan los tickets de autorización de kerberos que se activan al ejecutar **kinit**. Para lograrlo, basta con utilizar el comando **kdestroy**.

```
root@ubuntu:/etc# msktutil -N -c -b 'CN=Computers' -s BBMASPACE/bbminspace.local
-k bbmkeytab.keytab --computer-name BBMUBUNTU --upn BBMUBUNTUSER --server WIN-
G9HU7TCV86N.bbminspace.local --user-creds-only
root@ubuntu:/etc# kdestroy
```

Por último, podemos comprobar que el equipo se ha unido correctamente al dominio usando la herramienta *Usuarios y equipos de Active Directory* en el *controlador de dominio de Windows Server*.



Ahora pasamos a esa segunda parte de la que hablaba al principio, comenzando con configurar SSSD. SSSD (*System Security Services Daemon*) es un conjunto de *daemons* que se encarga de administrar la autenticación y el acceso a los recursos remotos mediante *LDAP* y *Kerberos*, entre otros.

Gracias a SSSD, se puede iniciar sesión en el equipo utilizando los datos de autenticación de una cuenta del dominio en lugar de hacerlo con una cuenta local. Ya debe encontrarse instalado en el sistema, pero ahora debemos configurarlo para que se comporte del modo adecuado.

Lo primero será copiar el archivo *keytab* que se generó al unir el cliente *Ubuntu* al dominio, dentro de la carpeta adecuada para la configuración de *SSSD* (*/etc/sssd*):
 sudo mv bbmkeytab.keytab /etc/sssd/bbmkeytab.keytab.

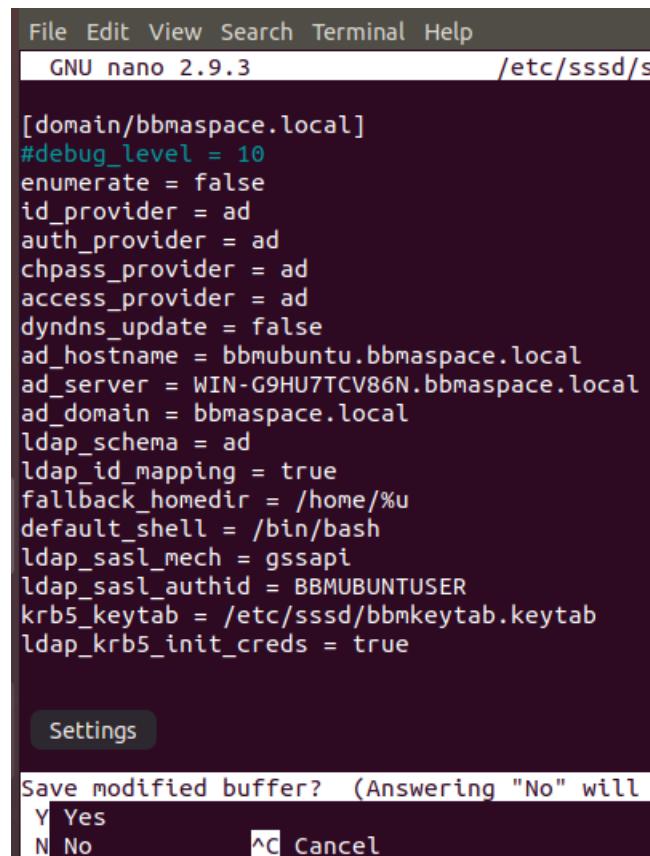
```
bbm@ubuntu:/etc$ sudo mv bbmkeytab.keytab /etc/sssd/bbmkeytab.keytab
bbm@ubuntu:/etc$
```

A diferencia de *Kerberos*, para *SSSD* no hay un archivo de configuración predeterminado. En este caso, editamos con: sudo nano /etc/sssd/sssd.conf.

```
bbm@ubuntu:/etc$ sudo nano /etc/sssd/sssd.conf
```

A continuación, copiaremos las siguientes líneas en el área de trabajo del editor... Pero no olvides modificar los valores que te indicamos en color rojo, para adaptarlos a los datos concretos de tu instalación:

```
[domain/bbminspace.local]
#debug_level = 10
enumerate = false
id_provider = ad
auth_provider = ad
chpass_provider = ad
access_provider = ad
dyndns_update = false
ad_hostname =
bbmubuntu.bbminspace.local
ad_server = WIN-
```



```
G9HU7TCV86N.bbminspace.local
ad_domain = bbminspace.local
ldap_schema = ad
ldap_id_mapping = true
fallback_homedir = /home/%u
```

```
default_shell = /bin/bash

ldap_sasl_mech = gssapi

ldap_sasl_authid = BBMUBUNTUSER

krb5_keytab = /etc/sssd/bbmkeytab.keytab
ldap_krb5_init_creds = true
```

NOTA: Cuando estemos listos, pulsamos la combinación de teclas **Ctrl** + **X** para cerrar el editor de textos. Intro para guardar.

Antes de abandonar, hay que asegurar que sus permisos son los adecuados. El propietario es el único que debe tener permisos de lectura y escritura sobre el archivo, y para lograrlo, sólo hay que escribir lo siguiente: `sudo chmod 0600 /etc/sssd/sssd.conf`.

```
bbm@ubuntu:/etc$ sudo nano /etc/sssd/sssd.conf
bbm@ubuntu:/etc$ sudo chmod 0600 /etc/sssd/sssd.conf
```

Se comprueba que el cambio ha surtido efecto recurriendo al comando ls: `sudo ls -al /etc/sssd`.

```
bbm@ubuntu:/etc$ sudo ls -al /etc/sssd
total 28
drwx--x--x  3 sssd sssd  4096 Jan  3  05:06 .
drwxr-xr-x 126 root root 12288 Jan  3  04:59 ..
-rw-------  1 root root  3527 Jan  3  04:43 bbmkeytab.keytab
drwxr-xr-x  2 root root  4096 May 11  2020 conf.d
-rw-------  1 root root   647 Jan  3  05:06 sssd.conf
bbm@ubuntu:/etc$
```

Ahora toca ajustar el comportamiento de **PAM (Pluggable Authentication Modules)** para iniciar sesión con usuarios del dominio. Es un *framework* que ofrece una capa de abstracción para diferentes métodos de autenticación (contraseña, tarjeta, parámetros biométricos, etc,).

En *Ubuntu*, uno de los archivos de configuración de *PAM* es **common-session**, que contiene parámetros generales sobre el inicio de sesión. Entre ellos, la necesidad de crear un directorio de inicio para las cuentas de usuario. Con esto se consigue que para las cuentas que pertenecen al dominio, añadiendo el argumento **pam_mkdirhome.so** en el archivo *common-session*. Al hacerlo, indicamos que se cree el directorio local, para cada cuenta, la primera vez que se inicie sesión con ella.

Para lograrlo, se edita el archivo de configuración: `sudo nano /etc/pam.d/common-session`.

```
bbm@ubuntu:/etc$ ^C
bbm@ubuntu:/etc$ sudo nano /etc/pam.d/common-session
```

Cuando se haya abierto el editor de textos, buscamos la línea session required pam_unix.so y debajo se escribe session required pam_mkhomedir.so skel=/etc/skel umask=0077.

```

File Edit View Search Terminal Help
GNU nano 2.9.3          /etc/pam.d/common-session           Modified
# pam-auth-update(8) for details.

# here are the per-package modules (the "Primary" block)
session [default=1]          pam_permit.so
# here's the fallback if no module succeeds
session requisite            pam_deny.so
# prime the stack with a positive return value if there isn't one already;
# this avoids us returning an error just because nothing sets a success code
# since the modules above will each just jump around
session required              pam_permit.so
# The pam_umask module will set the umask according to the system default in
# /etc/login.defs and user settings, solving the problem of different
# umask settings with different shells, display managers, remote sessions etc.
# See "man pam_umask".
session optional              pam_umask.so
# and here are more per-package modules (the "Additional" block)
session required              pam_unix.so
session required pam_mkhomedir.so skel=/etc/skel umask=0077
session optional              pam_sss.so
session optional              pam_systemd.so
# end of pam-auth-update config

Save modified buffer? (Answering "No" will DISCARD changes.)
Y Yes
N No      ^C Cancel

```

Para que se apliquen los cambios, hay que reiniciar el servicio sssd: sudo systemctl restart sssd.

```

bbm@ubuntu:/etc$ sudo systemctl restart sssd

```

Ahora toca convertir al administrador del dominio en administrador local. Si vamos a iniciar sesión con la cuenta *Administrador* del dominio de forma local en el equipo cliente, parece lógico que esta cuenta también tenga privilegios administrativos en el propio equipo. Para lograrlo, hay que incluir dicha cuenta en el grupo sudo, para lo que podemos utilizar la siguiente orden: sudo adduser administrator sudo.

```

bbm@ubuntu:/etc$ sudo adduser administrator sudo
Adding user `administrator' to group `sudo' ...
Adding user administrator to group sudo
Done.

```

A continuación, se comprueba que me puedo autenticar con el usuario Administrador (recuerda que esta cuenta no existe localmente y que se trata de una cuenta del dominio). Para lograrlo, basta con escribir la orden: su -l administrator.

```
administrator@ubuntu:~$ su -l administrator
No passwd entry for user 'administrator'
administrator@ubuntu:~$ cd
administrator@ubuntu:~$ sudo su
[sudo] password for administrator:
root@ubuntu:/home/administrator#
```

NOTA: ESTA CAPTURA SE ME PASÓ HACERLA, DEJO OTRA CON EL COMANDO Y EL DIRECTORIO CREADO.

Hago un reinicio con el comando `sudo reboot` para que se hagan efectivos los cambios:

```
examples.desktop
bbm@ubuntu:/home/administrator$ sudo reboot
```

NOTA: Ni siguiera tenemos que abandonar la cuenta *Administrator* ya que ahora forma parte del grupo *sudo*.

Al completar el reinicio, la interfaz gráfica muestra únicamente las cuentas de usuario que conoce. La cuenta de dominio que hay que usar es *Administrator*.



Inmediatamente veremos cómo se inicia sesión en la interfaz gráfica. Como es la primera vez que iniciamos sesión gráfica con esa cuenta, nos aparecerá el asistente de bienvenida.



Instalar SQL Server 2017 / 2019

Los siguientes comandos para SQL Server 2019 apuntan al repositorio de Ubuntu 18.04. Para configurar SQL Server en Ubuntu, se ejecutan los siguientes comandos en un terminal para instalar el paquete mssql-server.

Importar las claves de GPG del repositorio público:

```
wget -qO- https://packages.microsoft.com/keys/microsoft.asc | sudo apt-key add -
```

```
bbm@ubuntu:~$ wget -qO- https://packages.microsoft.com/keys/microsoft.asc | sudo apt-key add -
[sudo] password for bbm:
OK
```

Registraremos el repositorio de Ubuntu de Microsoft SQL Server para SQL Server 2019:

```
sudo add-apt-repository "$ (wget -qO-
https://packages.microsoft.com/config/ubuntu/18.04/mssql-server-
2019.list)"
```

```
bbm@ubuntu:~$ sudo add-apt-repository "$ (wget -qO- https://packages.microsoft.c
om/config/ubuntu/18.04/mssql-server-2019.list)"
Get:1 http://security.ubuntu.com/ubuntu bionic-security InRelease [88.7 kB]
Hit:2 http://us.archive.ubuntu.com/ubuntu bionic InRelease
Hit:3 http://us.archive.ubuntu.com/ubuntu bionic-updates InRelease
Hit:4 http://us.archive.ubuntu.com/ubuntu bionic-backports InRelease
Get:5 https://packages.microsoft.com/ubuntu/18.04/mssql-server-2019 bionic InRe
lease [10.5 kB]
Get:6 http://security.ubuntu.com/ubuntu bionic-security/main amd64 DEP-11 Metad
ata [48.9 kB]
Get:7 http://security.ubuntu.com/ubuntu bionic-security/universe amd64 DEP-11 M
etadata [59.5 kB]
Get:8 http://security.ubuntu.com/ubuntu bionic-security/multiverse amd64 DEP-11
Metadata [2,464 B]
Get:9 https://packages.microsoft.com/ubuntu/18.04/mssql-server-2019 bionic/main
amd64 Packages [13.4 kB]
Get:10 https://packages.microsoft.com/ubuntu/18.04/mssql-server-2019 bionic/mai
n arm64 Packages [2,713 B]
Get:11 https://packages.microsoft.com/ubuntu/18.04/mssql-server-2019 bionic/mai
n armhf Packages [2,713 B]
Fetched 229 kB in 2s (105 kB/s)
Reading package lists... Done
```

Ejecute los comandos siguientes para instalar SQL Server:

```
sudo apt-get update
```

```
bbm@ubuntu:~$ sudo apt-get update
Hit:1 http://security.ubuntu.com/ubuntu bionic-security InRelease
Hit:2 http://us.archive.ubuntu.com/ubuntu bionic InRelease
Hit:3 http://us.archive.ubuntu.com/ubuntu bionic-updates InRelease
Hit:4 http://us.archive.ubuntu.com/ubuntu bionic-backports InRelease
Hit:5 https://packages.microsoft.com/ubuntu/18.04/mssql-server-2019 bionic InRe
lease
Reading package lists... Done
```

```
sudo apt-get install -y mssql-server
```

```
bbm@ubuntu:~$ sudo apt-get install -y mssql-server
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
+-----+
Please run 'sudo /opt/mssql/bin/mssql-conf setup'
to complete the setup of Microsoft SQL Server
+-----+
Processing triggers for gnome-menus (3.13.3-11ubuntu1.1) ...
Processing triggers for mime-support (3.60ubuntu1) ...
Processing triggers for desktop-file-utils (0.23-1ubuntu3.18.04.2) ...
Processing triggers for libc-bin (2.27-3ubuntu1.2) ...
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
bbm@ubuntu:~$
```

Cuando finaliza la instalación del paquete, ejecutamos lo que pide en el anterior punto **mssql-conf setup** y seguimos las indicaciones para establecer la contraseña de administrador del sistema y elegir la edición.

```
sudo /opt/mssql/bin/mssql-conf setup
```

```
bbm@ubuntu:~$ sudo /opt/mssql/bin/mssql-conf setup
usermod: no changes
Choose an edition of SQL Server:
  1) Evaluation (free, no production use rights, 180-day limit)
  2) Developer (free, no production use rights)
  3) Express (free)
  4) Web (PAID)
  5) Standard (PAID)
  6) Enterprise (PAID) - CPU Core utilization restricted to 20 physical/40 hyperthreaded
  7) Enterprise Core (PAID) - CPU Core utilization up to Operating System Maximum
  8) I bought a license through a retail sales channel and have a product key to enter.

Enter your edition(1-8): 2
The license terms for this product can be found in
/usr/share/doc/mssql-server or downloaded from:
https://go.microsoft.com/fwlink/?LinkId=2104294&clcid=0x409

The privacy statement can be viewed at:
https://go.microsoft.com/fwlink/?LinkId=853010&clcid=0x409

Do you accept the license terms? [Yes/No]:Yes

Enter the SQL Server system administrator password:
Confirm the SQL Server system administrator password:
Configuring SQL Server...

ForceFlush is enabled for this instance.
ForceFlush feature is enabled for log durability.
Created symlink /etc/systemd/system/multi-user.target.wants/mssql-server.service → /lib/systemd/system/mssql-server.service.
Setup has completed successfully. SQL Server is now starting.
bbm@ubuntu:~$
```

Cuando finaliza la configuración, se comprueba que el servicio se esté ejecutando:

```
systemctl status mssql-server --no-pager
```

```
bbm@ubuntu:~$ systemctl status mssql-server --no-pager
● mssql-server.service - Microsoft SQL Server Database Engine
   Loaded: loaded (/lib/systemd/system/mssql-server.service; enabled; vendor pr
eset: enabled)
     Active: active (running) since Tue 2020-11-03 11:28:06 PST; 37s ago
       Docs: https://docs.microsoft.com/en-us/sql/linux
   Main PID: 8548 (sqlservr)
     Tasks: 122
    CGroup: /system.slice/mssql-server.service
              └─8548 /opt/mssql/bin/sqlservr
                  ├─8569 /opt/mssql/bin/sqlservr

Nov 03 11:28:25 ubuntu sqlservr[8548]: [61B blob data]
Nov 03 11:28:29 ubuntu sqlservr[8548]: [96B blob data]
Nov 03 11:28:29 ubuntu sqlservr[8548]: [66B blob data]
Nov 03 11:28:29 ubuntu sqlservr[8548]: [75B blob data]
Nov 03 11:28:29 ubuntu sqlservr[8548]: [96B blob data]
Nov 03 11:28:29 ubuntu sqlservr[8548]: [100B blob data]
Nov 03 11:28:29 ubuntu sqlservr[8548]: [71B blob data]
Nov 03 11:28:30 ubuntu sqlservr[8548]: [124B blob data]
Nov 03 11:28:30 ubuntu sqlservr[8548]: [73B blob data]
Nov 03 11:28:30 ubuntu sqlservr[8548]: [186B blob data]
```

NOTA: Si planeamos conectarnos de forma remota, es posible que también haya que abrir el puerto TCP de SQL Server (valor predeterminado: 1433) en el firewall.

En este momento, SQL Server 2019 se está ejecutando en el equipo Ubuntu y está listo para usarse.

Instalar las herramientas de línea de comandos de SQL Server

Para crear una base de datos, necesita conectarse con una herramienta que pueda ejecutar instrucciones Transact-SQL en SQL Server. En los pasos siguientes, se instalan las herramientas de línea de comandos de SQL Server sqlcmd y bcp.

Siga estos pasos para instalar mssql-tools en Ubuntu.

Importe las claves de GPG del repositorio público.

Bash

Copiar

```
curl https://packages.microsoft.com/keys/microsoft.asc | sudo apt-key add -
```

Registre el repositorio de Ubuntu de Microsoft.

Bash

Copiar

```
curl https://packages.microsoft.com/config/ubuntu/18.04/prod.list | sudo tee /etc/apt/sources.list.d/msprod.list
```

Actualice la lista de orígenes y ejecute el comando de instalación con el paquete para desarrolladores de unixODBC. Para más información, consulte Instalación de Microsoft ODBC Driver for SQL Server (Linux).

Bash

Copiar

```
sudo apt-get update
```

```
sudo apt-get install mssql-tools unixodbc-dev
```

Nota

Para actualizar a la versión más reciente de mssql-tools, ejecute los siguientes comandos:

Bash

Copiar

```
sudo apt-get update
```

```
sudo apt-get install mssql-tools
```

Opcional: agregue /opt/mssql-tools/bin/ a la variable de entorno PATH en un shell de Bash.

Para que sqlcmd/bcp sea accesible desde el shell de Bash para inicios de sesión, modifique la variable PATH en el archivo ~/.bash_profile con el comando siguiente:

Bash

Copiar

```
echo 'export PATH="$PATH:/opt/mssql-tools/bin"' >> ~/.bash_profile
```

Para que sqlcmd/bcp sea accesible desde el shell de Bash para sesiones interactivas o que no sean de inicio de sesión, modifique la variable PATH en el archivo ~/.bashrc con el comando siguiente:

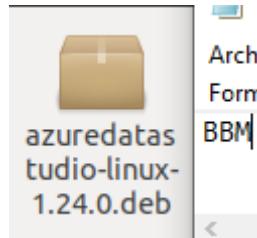
Bash

Copiar

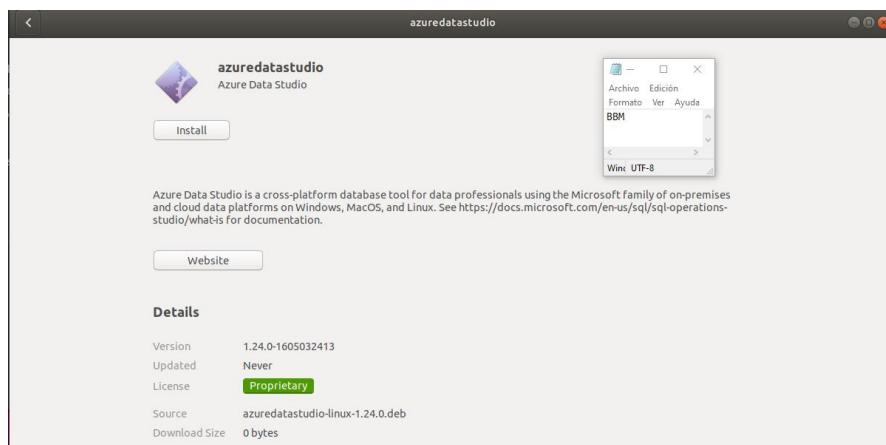
```
echo 'export PATH="$PATH:/opt/mssql-tools/bin"' >> ~/.bashrc
source ~/.bashrc
```

Instalar GUI: Azure Data Studio

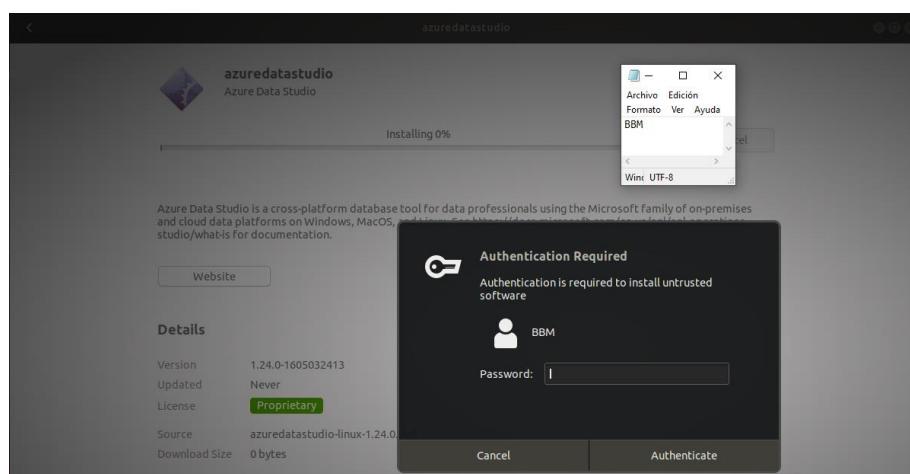
Descargamos Azure Data Studio para Linux en el enlace <https://docs.microsoft.com/es-es/sql/azure-data-studio/download-azure-data-studio?view=sql-server-ver15>, mediante uno de los instaladores, o bien el archivo tar.gz. Yo descargué la versión .deb.



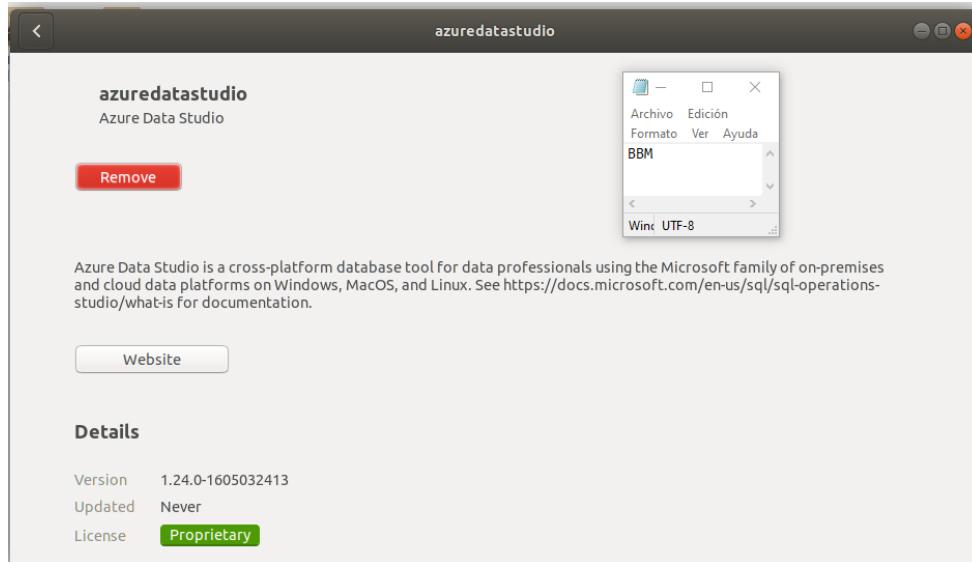
Vamos a la ruta y hacemos doble clic sobre el archivo descargado. Se abre una nueva ventana:



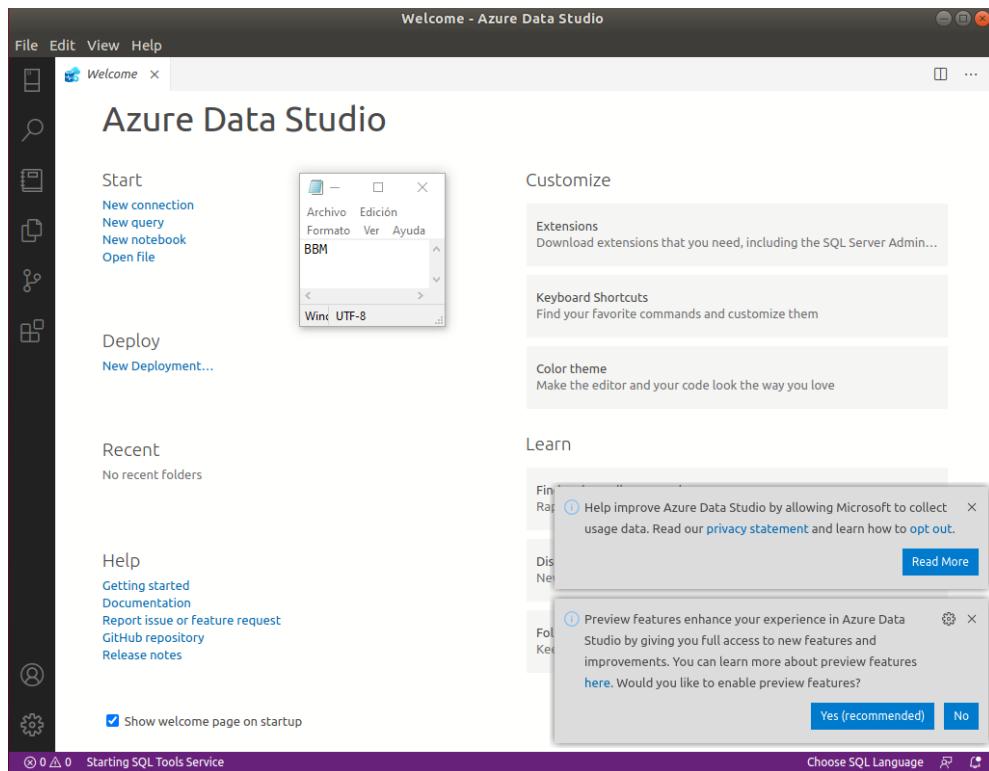
Se abre una ventana en la que nos pide la autenticación, y ya comienza a cargar la instalación:



Una vez finalice, no se ve ningún tipo de mensaje, simplemente se ve la misma ventana de instalación con un mensaje que pone Remove:



Por último, buscamos la aplicación (ícono) y vemos la interfaz gráfica:

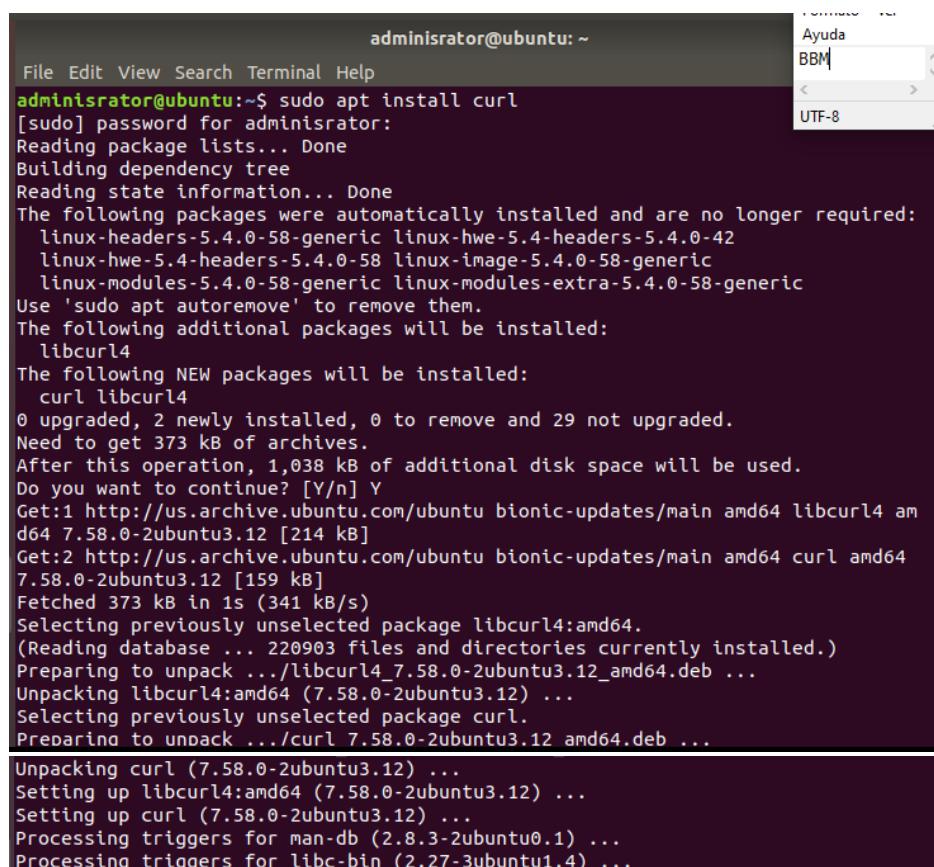


Instalación y uso de Sqlcmd

La utilidad **sqlcmd** permite escribir instrucciones Transact-SQL, procedimientos del sistema y archivos de script a través de diversos modos disponibles:

- En el símbolo del sistema.
- En el modo SQLCMD del **Editor de consultas**.
- En un archivo de script de Windows.
- En un paso de trabajo del sistema operativo (cmd.exe) de un trabajo del Agente SQL Server.

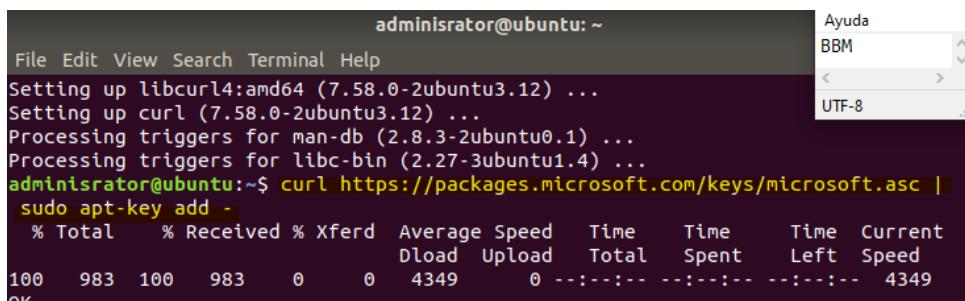
Para su instalación requiere **curl**. Para ello usamos el comando **sudo apt install curl**.



```
administrator@ubuntu:~$ sudo apt install curl
[sudo] password for administrator:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  linux-headers-5.4.0-58-generic linux-hwe-5.4-headers-5.4.0-42
  linux-hwe-5.4-headers-5.4.0-58 linux-image-5.4.0-58-generic
  linux-modules-5.4.0-58-generic linux-modules-extra-5.4.0-58-generic
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  libcurl4
The following NEW packages will be installed:
  curl libcurl4
0 upgraded, 2 newly installed, 0 to remove and 29 not upgraded.
Need to get 373 kB of archives.
After this operation, 1,038 kB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://us.archive.ubuntu.com/ubuntu bionic-updates/main amd64 libcurl4 amd64 7.58.0-2ubuntu3.12 [214 kB]
Get:2 http://us.archive.ubuntu.com/ubuntu bionic-updates/main amd64 curl amd64 7.58.0-2ubuntu3.12 [159 kB]
Fetched 373 kB in 1s (341 kB/s)
Selecting previously unselected package libcurl4:amd64.
(Reading database ... 220903 files and directories currently installed.)
Preparing to unpack .../libcurl4_7.58.0-2ubuntu3.12_amd64.deb ...
Unpacking libcurl4:amd64 (7.58.0-2ubuntu3.12) ...
Selecting previously unselected package curl.
Preparing to unpack .../curl_7.58.0-2ubuntu3.12_amd64.deb ...
Unpacking curl (7.58.0-2ubuntu3.12) ...
Setting up libcurl4:amd64 (7.58.0-2ubuntu3.12) ...
Setting up curl (7.58.0-2ubuntu3.12) ...
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
Processing triggers for libc-bin (2.27-3ubuntu1.4) ...
```

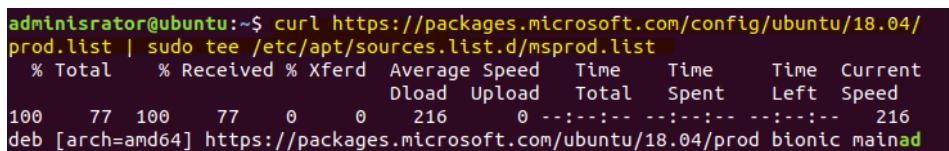
Importamos el repositorio público GPG Keys:

curl https://packages.microsoft.com/keys/microsoft.asc | sudo apt-key add - es el comando que se usa.



```
administrator@ubuntu:~$ curl https://packages.microsoft.com/keys/microsoft.asc | sudo apt-key add -
% Total    % Received % Xferd  Average Speed   Time   Time     Time  Current
          Dload  Upload Total Spent   Left Speed
100  983  100  983    0      0  4349      0 --:--:-- --:--:-- --:--:--  4349
OK
```

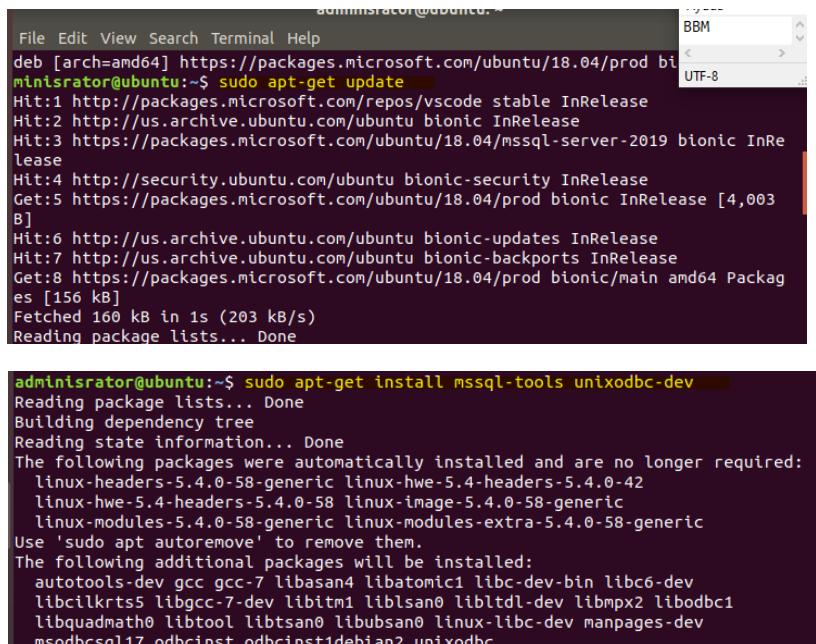
Registraremos también el repositorio Microsoft Ubuntu (curl https://packages.microsoft.com/config/ubuntu/18.04/prod.list | sudo tee /etc/apt/sources.list.d/msprod.list)



```
administrator@ubuntu:~$ curl https://packages.microsoft.com/config/ubuntu/18.04/
prod.list | sudo tee /etc/apt/sources.list.d/msprod.list
% Total    % Received % Xferd  Average Speed   Time   Time     Time  Current
          Dload  Upload Total Spent   Left Speed
100  77  100  77    0      0  216      0 --:--:-- --:--:--  216
deb [arch=amd64] https://packages.microsoft.com/ubuntu/18.04/prod bionic mainad
```

NOTA: Esto lo estamos instando en la versión 18.04. Para otras versiones, el directorio de instalación cambia.

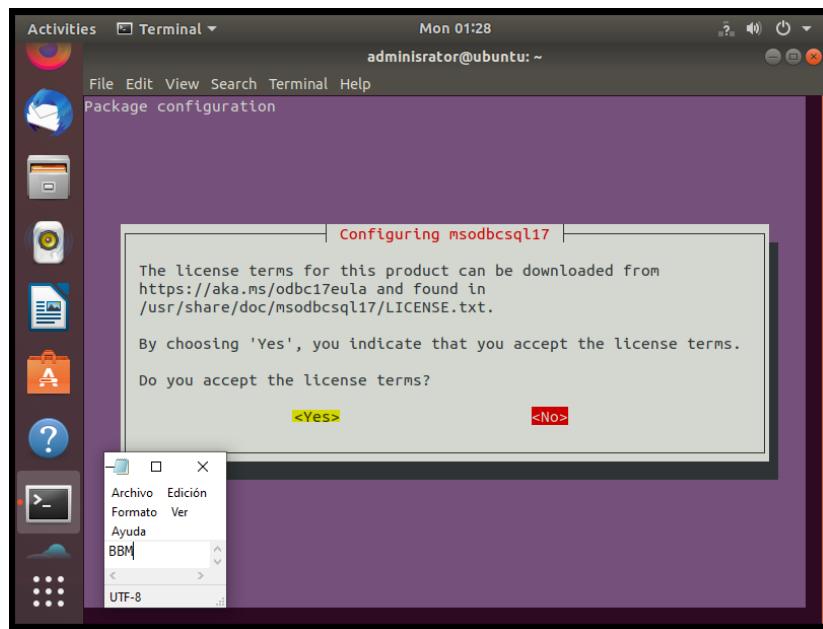
Actualizamos y arrancamos la instalación con el unixODBC developer package: sudo apt-get update y sudo apt-get install mssql-tools unixodbc-dev.



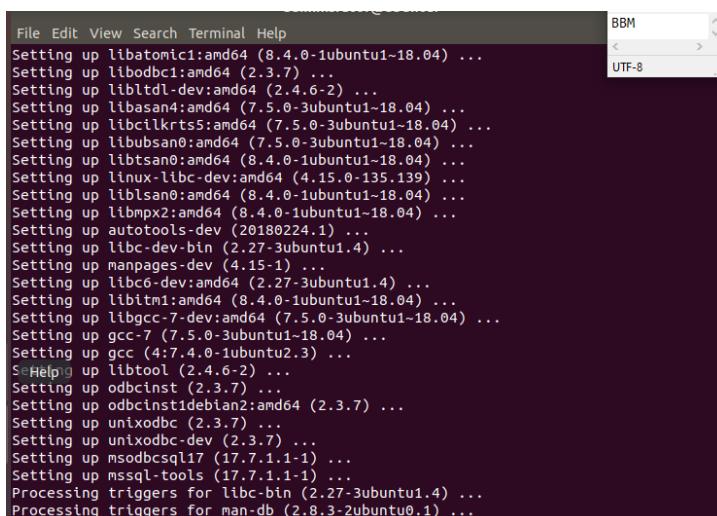
```
administrator@ubuntu:~$ sudo apt-get update
Hit:1 http://packages.microsoft.com/repos/vscode stable InRelease
Hit:2 http://us.archive.ubuntu.com/ubuntu bionic InRelease
Hit:3 https://packages.microsoft.com/ubuntu/18.04/mssql-server-2019 bionic InRelease
Hit:4 http://security.ubuntu.com/ubuntu bionic-security InRelease
Get:5 https://packages.microsoft.com/ubuntu/18.04/prod bionic InRelease [4,003 B]
Hit:6 http://us.archive.ubuntu.com/ubuntu bionic-updates InRelease
Hit:7 http://us.archive.ubuntu.com/ubuntu bionic-backports InRelease
Get:8 https://packages.microsoft.com/ubuntu/18.04/prod bionic/main amd64 Packages [156 kB]
Fetched 160 kB in 1s (203 kB/s)
Reading package lists... Done

administrator@ubuntu:~$ sudo apt-get install mssql-tools unixodbc-dev
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  linux-headers-5.4.0-58-generic linux-hwe-5.4-headers-5.4.0-42
  linux-hwe-5.4-headers-5.4.0-58 linux-image-5.4.0-58-generic
  linux-modules-5.4.0-58-generic linux-modules-extra-5.4.0-58-generic
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  autotools-dev gcc gcc-7 libasan4 libatomic1 libc-dev-bin libc6-dev
  libcilkrt5 libgcc-7-dev libitm1 libltsan0 libltdl-dev libmpx2 libodbc1
  libquadmath0 libtool libtsan0 libubsan0 linux-libc-dev manpages-dev
  msodbcsql17 odbcinst odbcinst1debian2 unixodbc
```

Cuando sale el mensaje de aceptar los términos de licencia le damos a Yes.

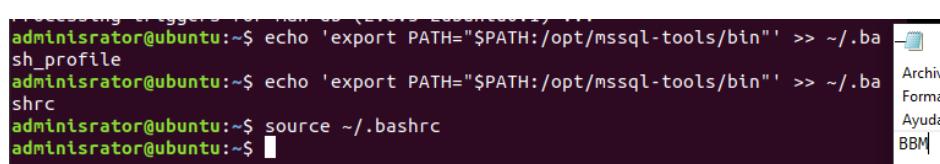


A partir de aquí empieza a cargar la instalación.

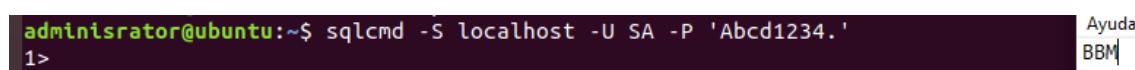


Una vez termina, añadimos las rutas para hacerlo accesible desde la Shell para los logines: echo 'export PATH="\$PATH:/opt/mssql-tools/bin"' >> ~/.bash_profile.

Para las no-login sessions metemos el siguiente: echo 'export PATH="\$PATH:/opt/mssql-tools/bin"' >> ~/.bashrc source ~/.bashrc.



Para conectarnos a SQL Server de manera local, aplicamos el comando `sqlcmd -S localhost -U SA -P <YourPassword>`.



Una vez conectado, se ve un símbolo como este: 1>. Como primer paso hacemos @@version para ver la versión que hemos instalado y comprobar el funcionamiento.

```
Incorrect syntax near 'SELECT'.
1> SELECT @@version
2> GO

Microsoft SQL Server 2019 (RTM-CU8) (KB4577194) - 15.0.4073.23 (X64)
Sep 23 2020 16:03:08
Copyright (C) 2019 Microsoft Corporation
Developer Edition (64-bit) on Linux (Ubuntu 18.04.5 LTS) <X64>

(1 rows affected)
1> 
```

Para conectarse a una instancia remota, se cambia en el comando de conexión localhost por el nombre o la dirección IP (como ya está todo conectado en el dominio, no hace falta hacer nada más)

```
Sqlcmd: Error: Microsoft ODBC Driver 17 for SQL Server : Cannot generate SSPI context.
administrator@ubuntu:~$ sqlcmd -S 192.168.0.3 -Usa -Psa
1> 
```

Ahora voy a hacer un para de pruebas en la conexión para comprobar el funcionamiento de sqlcmd:

- Cómo verificar la base de datos actual

En el momento en que se crea un inicio de sesión de SQL Server, se puede definir la base de datos predeterminada en la que se desea iniciar sesión. Si no se especifica, la base de datos maestra esta es la predeterminada

```
select DB_NAME()
GO
```

```
1> SELECT DB_NAME()
2> GO

-----
master

(1 rows affected)
1> 
```

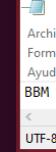
- Cómo enumerar las bases de datos

Las siguientes oraciones enumerarán las bases de datos en la instancia de SQL:

```
select name from sys.databases
go
```

En la tabla sys.databases, tiene toda la información de la base de datos:

```
1> SELECT NAME FROM sys.databases
2> GO
NAME
-----
master
tempdb
model
msdb
NORTHWND
pubs
AdventureWorks2017
WideWorldImporters
QUECOVID
QUECOVID2
TestDB
BBM_ASPACE
SQLFileTable
PruebaFS
Colegio_Medico
CAR
TRIALBBM
(17 rows affected)
1>
```

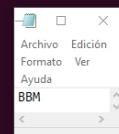


También puede usar el procedimiento almacenado sp_databases:

Sp_databases

Go

```
1> sp_databases
2> go
DATABASE_NAME          DATABASE_SIZE REMARKS
-----
AdventureWorks2017           344064 NULL
BBM_ASPACE                   30720 NULL
CAR                           16384 NULL
Colegio_Medico                40896 NULL
master                         7296 NULL
model                          16384 NULL
msdb                           21760 NULL
NORTHWND                       9024 NULL
PruebaFS                      16384 NULL
pubs                           16384 NULL
QUECOVID                       16384 NULL
QUECOVID2                      16384 NULL
SQLFileTable                   16384 NULL
tempdb                          24576 NULL
TestDB                          16384 NULL
TRIALBBM                        40896 NULL
WideWorldImporters              799104 NULL
```



Para demostrar el uso, vamos a hacer un script sencillo

--ACTIVAMOS LA BASE DE DATOS

```
1> use trialbbm  
2> go
```

```
1> use trialbbm  
2> go  
Changed database context to 'TRIALBBM'.
```

--VEMOS LAS COLUMNAS QUE TIENE DENTRO

```
1> select name from sysobjects where type='U'  
2> go
```

```
1> select name from sysobjects where type='U'  
2> go  
name
```

```
-----  
-----  
BBM_Expediente  
  
BBM_Sesion  
  
BBM_Comidas  
  
BBM_Enfermedad_Diagnosticada  
  
BBM_Tratamientos  
  
BBM_Especialista  
  
BBM_Usuario  
  
SWITCH_BBM  
  
(8 rows affected)
```

-- CREAMOS UNA BASE DE DATOS

```
1> DROP DATABASE IF EXISTS BBM_SQLCMDUBUNTU  
2> GO  
1> CREATE DATABASE BBM_SQLCMDUBUNTU  
2> GO  
1> USE BBM_SQLCMDUBUNTU  
2> GO
```

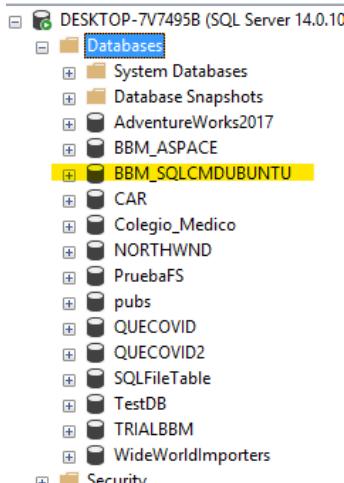
Changed database context to 'BBM_SQLCMDUBUNTU'.

```
1> DROP DATABASE IF EXISTS BBM_SQLCMDUBUNTU  
2> GO  
1> CREATE DATABASE BBM_SQLCMDUBUNTU  
2> GO  
1> USE BBM_SQLCMDUBUNTU  
2> GO  
Changed database context to 'BBM_SQLCMDUBUNTU'.
```

--CREAMOS UNA TABLA

```
1> DROP TABLE IF EXISTS BBM
2> GO
1> CREATE TABLE BBM (ID INT,NAME VARCHAR(20))
2> GO
1>
Changed database context to "BBM_SQLCMDUBUNTU".
1> DROP TABLE IF EXISTS BBM
2> GO
1> CREATE TABLE BBM (ID,NAME(
2> GO
Msg 173, Level 15, State 13, Server DESKTOP-7V7495B, Line 1
The definition for column 'ID' must include a data type.
1> CREATE TABLE BBM (ID INT,NAME VARCHAR(20))
2> GO
```

Refrescamos en la GUI de W10 y vemos como se crea la base de datos:



-- VEMOS LA TABLA CREADA EN LA BASE

```
1> Select name from sysobjects where type='U'
2> GO
name
```


BBM

```
1> Select name from sysobjects where type='U'
2> GO
name
```


BBM

```
(1 rows affected)
```

-- VEMOS EL CONTENIDO DE LA TABLA

```
1> SELECT * FROM BBM
```

```
2> GO
```

ID	NAME

```
(0 rows affected)
```

```
1>
```

```
1> SELECT * FROM BBM
2> GO
ID          NAME
-----
(0 rows affected)
1>
```

-- BORRAMOS LA BASE DE DATOS

```
1> USE master
```

```
2> GO
```

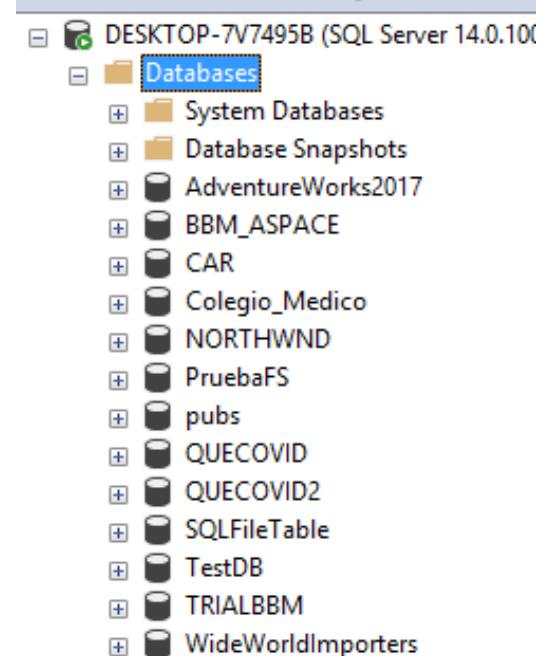
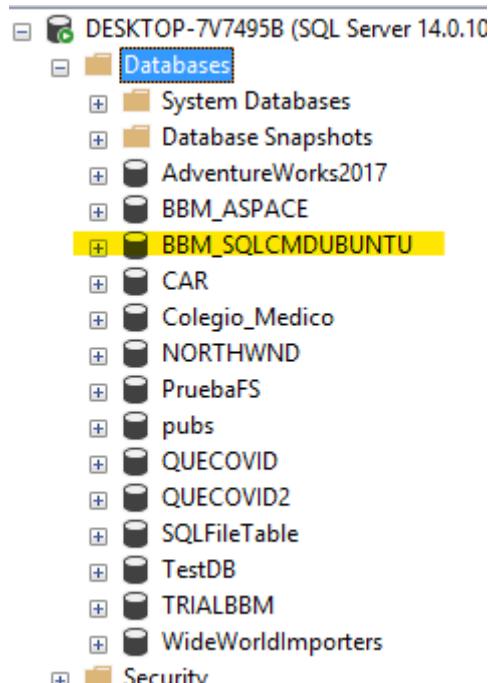
Changed database context to 'master'.

```
1> DROP DATABASE IF EXISTS BBM_SQLCMDUBUNTU
```

```
2> GO
```

```
1> USE master
2> GO
Changed database context to 'master'.
1> DROP DATABASE IF EXISTS BBM_SQLCMDUBUNTU
2> GO
```

Controlamos en la base de datos que se ha borrado



MODELADO BASE DE DATOS

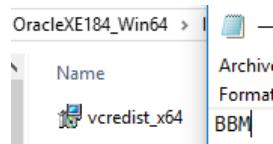
Modelado con SQL Data Modeler

Instalación de Oracle

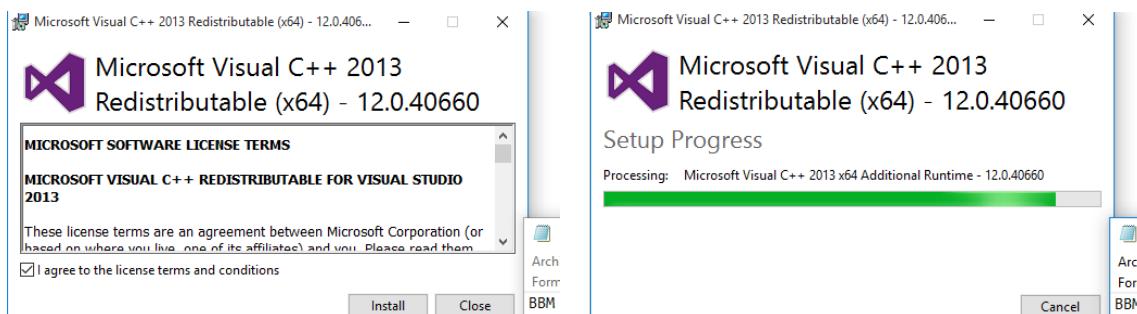
Para la instalación de Oracle, vamos a elegir la versión express en el siguiente enlace <https://www.oracle.com/database/technologies/xe-downloads.html>. De las dos opciones que nos encontramos, elegimos la versión para Windows:



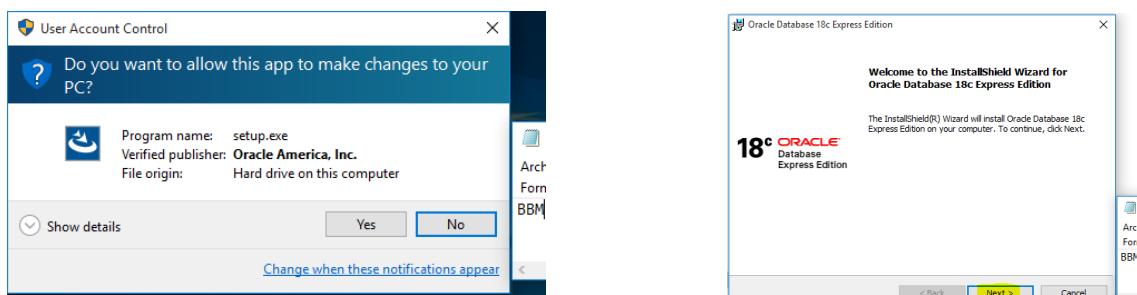
Una vez descargado, para evitar problemas en la instalación instalamos los **Microsoft Visual C++ 2013 Redistributable** (lo encontramos dentro del propio instalador en la ruta `C:\Users\BBM\Desktop\OracleXE184_Win64\ISSetupPrerequisites\{C445AD52-1E5D-47BA-841E-2E252A938AE7\}`).



Realmente son dos ventanas en las que clicamos **Install** y esperamos. Cuando acaba nos pide reiniciar; lo hacemos y finaliza esta parte:



Volviendo ya al *Express*, ejecutamos como administrador y comienza la instalación con la presentación del asistente (*Wizard*):

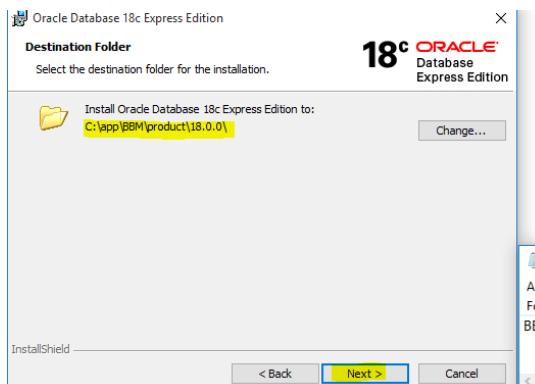


NOTA: A veces nos puede dar problemas a la hora de instalar y tendremos que volver a repetir el proceso (a mí personalmente me funcionó a la primera)

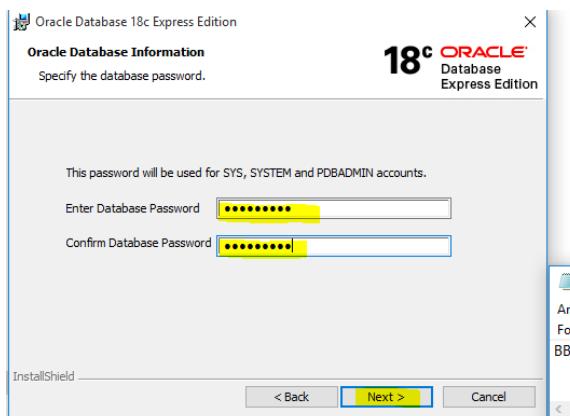
En la siguiente ventana acepto los términos:



En la siguiente ventana dice la ruta (`C:\app\BBM\product\18.0.0` → 18.0.0 es la versión de Oracle). Se deja por defecto porque al cambiarla suele dar problemas cambiarla de ruta (hay variables de entorno que apuntan a esa carpeta). Aún con todo, si hubiese problemas, dejaríamos la ruta por defecto en `C:/`, nunca en *Documentos* ni en el *Escritorio*.

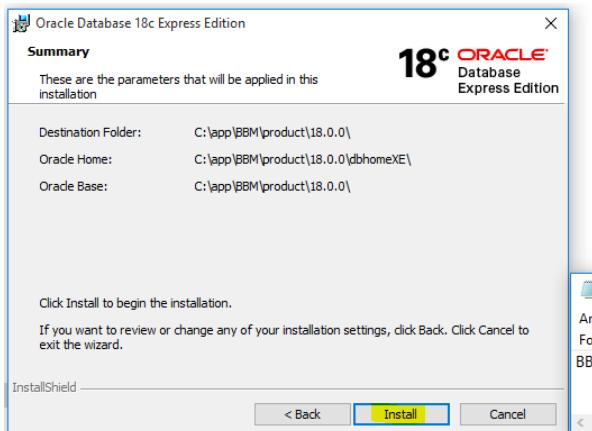


En la siguiente ventana me pide meter la contraseña. Dejaré como en todas *Abcd1234* :

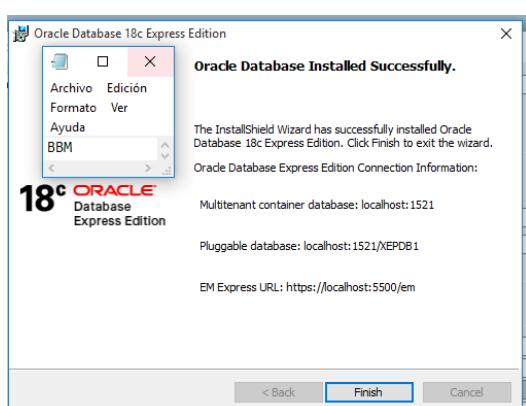
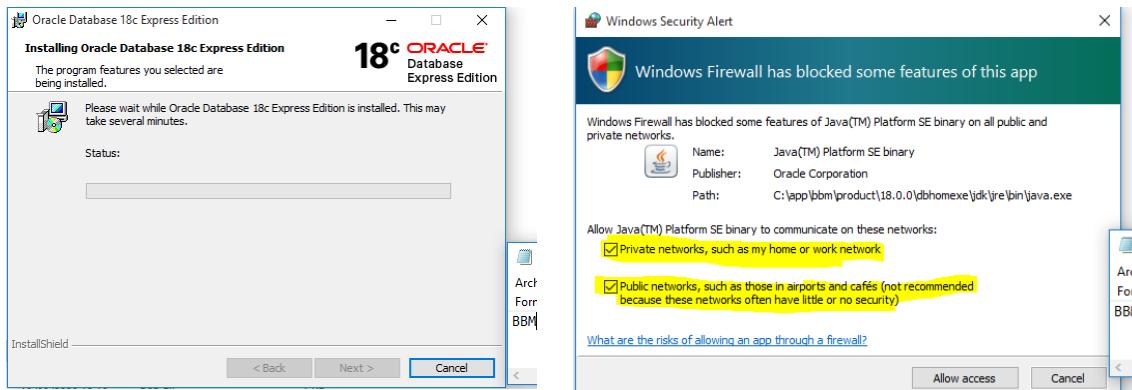


Una ventana muy importante es esta ventana, porque se ve donde está instalando todo, porque nos resultará mucho más sencillo saber dónde nos

va a dar un error de conexión. Por ejemplo, si nos da un fallo de *network*, sabremos buscar en la ruta de *Oracle Home*. Le damos a instalar:



Se pondrá una pantalla de carga y llegará un momento en el que nos preguntará una pantalla a modo de *pop-up* en el que se nos indicará que el *Firewall de Windows* nos quiere bloquear algunas funciones de Java, y las funciones si nos fijamos están apuntando sobre las rutas de la anterior ventana (por eso es bueno no cambiar nunca las rutas que nos vienen por defecto). En este caso activamos ambos *check* y permito el acceso.

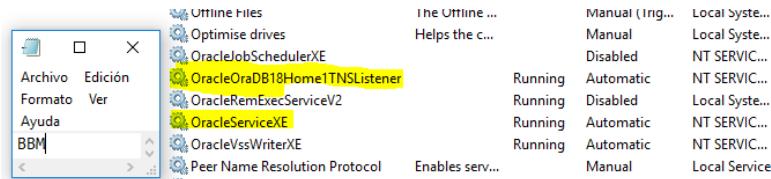


Para acabar nos sale la última ventana donde nos indica que nos sale correctamente. Nos da una cadena de conexión que en realidad actúa como un administrador desde el Browser para la base de datos, cosa que SQLServer no tiene.

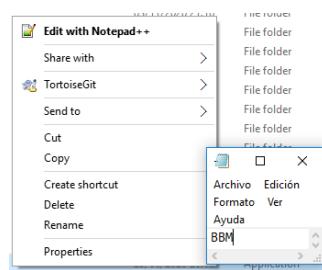
SQL Developer - Oracle Express

SQL Developer no necesita instalación, nada más descargarlo, lo ubico en C:/. Al entrar, ya encuentro el icono  **sqldeveloper**.

NOTA: Para que funcione Oracle, tenemos que tener dos servicios activos: *OracleOraDB18Home1TNSListener* (para escuchar) y *OracleServiceXE* (el equivalente a **mssqlserver**).

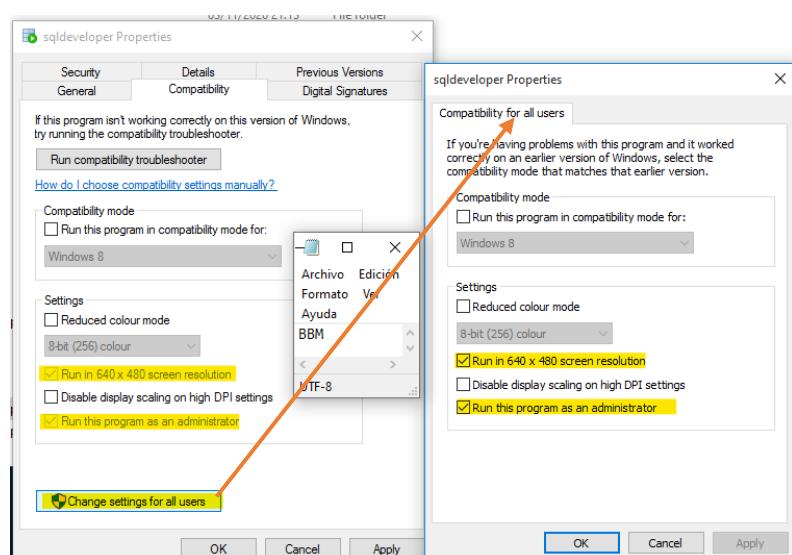


Si por lo que sea, necesito cambiar el tamaño del entorno gráfico, será haciendo botón derecho sobre el icono > Propiedades.



Seleccionamos la pestaña **Compatibility**, y en ella hacemos check a los dos *Run in 640x480 screen resolution* y en *Run this program as an administrator*.

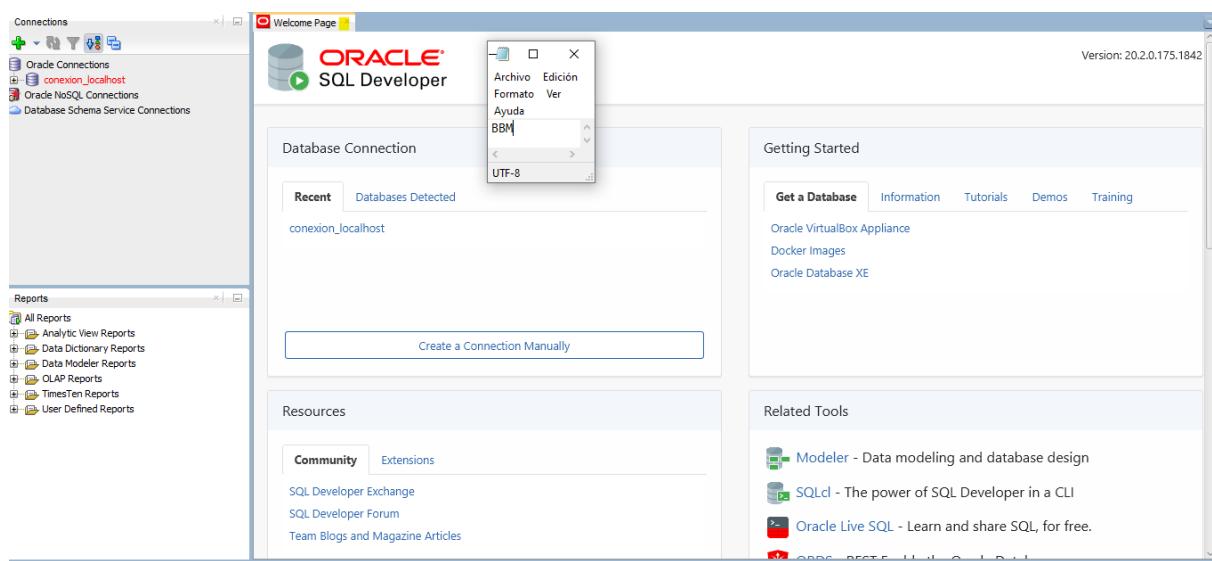
Hacemos clic en *Change high DPI settings*, y dentro hacemos check sobre *Override high DPI scaling behavior*. *Scaling performed by:*



Hacemos clic en *Change settings for all users* donde repito los dos pasos anteriores en la nueva ventana que se me abre.

NOTA: Todo esto lo he tenido que reproducir según la información proporcionada, pues no me aparecían algunas opciones

Una vez hecho esto, lanzo el programa como administrador. El entorno es muy parecido. Lo primero que hacemos es descartar la página de inicio.



Conexiones

Tenemos dos tipos de conexiones: La *conexión de Oracle* y la *conexión de sqlserver* para la cual necesitamos unos drivers de java.

A mayores, como está citado en el apartado anterior, tenemos que cerciorarnos que están operativos los servicios *xe home* y *oracletns listener*.

Vamos a probar a conectarnos a Oracle desde diferentes ámbitos → CMD y GUI:

Desde CMD

Lo primero que vamos a hacer es ejecutar **CMD** como administradores.

Una vez se abra, metemos **sqlplus**. Como hacen algunas bases de datos, ponen la ruta de los ejecutables en un *path* de lo que llamamos variables de entorno, lo cuál nos permite ejecutar desde cual ruta en vez de tener que ir a la ruta. EL programa de Oracle para SQL se llama **SQL*Plus**, y lo que voy a hacer es poner una cuenta como *SYSDBA* y cambiarme al usuario *XEPDB1*. También existe un usuario que es *Humanresources*, el cuál se desbloqueará más adelante.

```
Microsoft Windows [Version 10.0.10240]
(c) 2015 Microsoft Corporation. All rights reserved.
```

```
C:\Windows\system32>sqlplus → Ejecutado como administrador
```

```
SQL*Plus: Release 18.0.0.0.0 - Production on Mon Nov 9 12:51:13 2020
Version 18.4.0.0.0 → Hubo un salto de la 12 a la 18
```

```
Copyright (c) 1982, 2018, Oracle. All rights reserved.
```

```
Enter user-name: SYS / AS SYSDBA
Enter password:
```

El prompt que me aparece al logarme es SQL (esto me indica que ya estoy en Oracle). Oracle tiene una política de pocas bases de datos y otra que se llama *table space*, donde las bases de datos se “crean”. Con el comando **show user** mostramos el usuario actual que estamos usando.

```
Connected to:
Oracle Database 18c Express Edition Release 18.0.0.0.0 - Production
Version 18.4.0.0.0

SQL> SHOW USER
USER is "SYS"
```

Ahora nos vamos a cambiar de la base de datos **SYSDBA** a la base de datos **XEPDB1**. Con el comando `SELECT name, open_mode FROM v$pdbs;` podemos ver las bases de datos del sistema:

```
SQL> ALTER SESSION SET container = XEPDB1;
```

`Session altered.` → me confirma el cambio de base de datos

```
SQL> SELECT name, open_mode FROM v$pdbs;
```

NAME

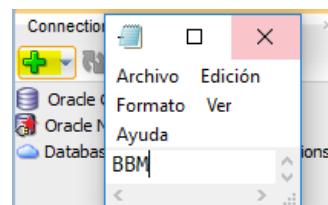
OPEN_MODE

XEPDB1
READ WRITE → tengo los permisos de lectura y escritura

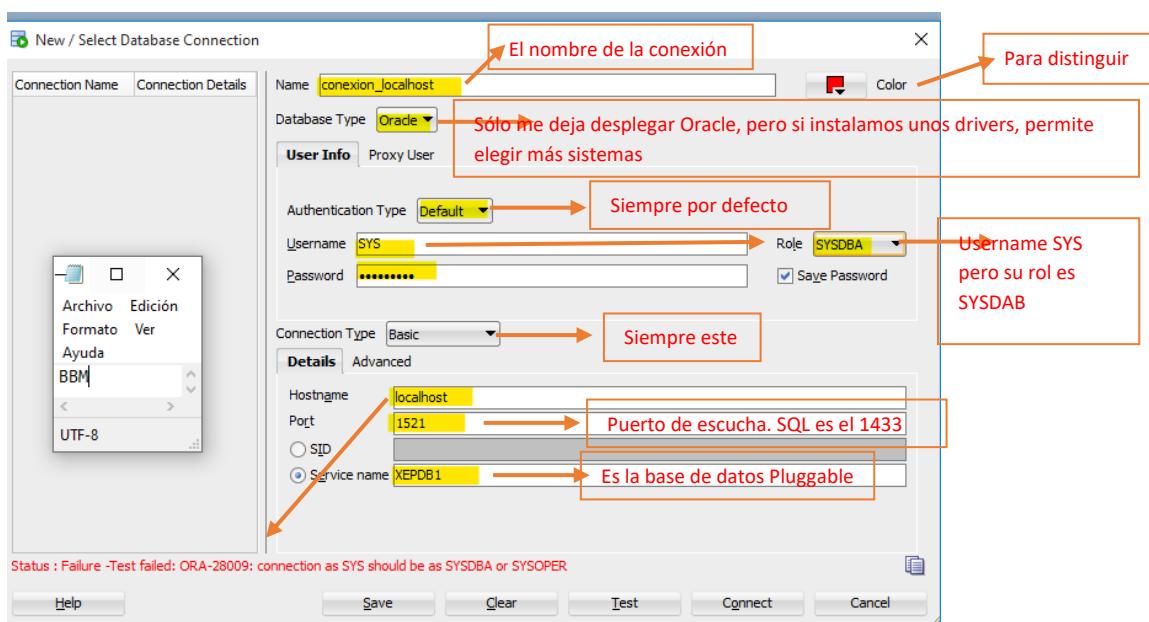
```
SQL> ALTER PLUGGABLE DATABASE open; → Da error porque ya está abierta
ALTER PLUGGABLE DATABASE open
*
ERROR at line 1:
ORA-65019: pluggable database XEPDB1 already open
```

`SQL>` → Me indica que estoy en Oracle
Desde GUI

En esta parte hay varios tipos de conexiones... SQL, Oracle, etc. Empezaré por la de **Oracle**... ¿Cómo nos conectamos? En este caso creamos una sesión.



Acto seguido nos sale la siguiente ventana que explicaré en el siguiente dibujo:



Al configurar todo, antes de darle a Connect, pulsamos **Test** para ver como si establece bien las conexiones. Aquí me dió un error porque la captura de pantalla me equivoqué en varios aspectos. El primero fue que no había elegido SYSDBA como rol (lo cuál ya aparece cambiado) y el segundo es un error muy común que es debido al *listener*. Para solventarlo lo que tenemos que hacer es ir a la ruta C:\app\BBM\product\18.0.0\dbhomeXE\network\admin. Allí nos encontramos con los archivos **listener.ora** y **tnsnames**. Hacemos un backup de los mismos y editamos los que tenemos cambiando la IP de HOST por el nombre localhost.

ANTES:

```
# listener.ora Network Configuration File: C:\app\BBM\product\18.0.0\dbhomeXE\NETWORK\LISTENER.ORA
# Generated by Oracle configuration tools.

DEFAULT_SERVICE_LISTENER = XE

SID_LIST_LISTENER =
  (SID_LIST =
    (SID_DESC =
      (SID_NAME = CLRExtProc)
      (ORACLE_HOME = C:\app\BBM\product\18.0.0\dbhomeXE)
      (PROGRAM = extproc)
      (ENVS = "EXTPROC_DLLS=ONLY:C:\app\BBM\product\18.0.0\dbhomeXE\bin\oraclr18.dll")
    )
  )

LISTENER =
  (DESCRIPTION_LIST =
    (DESCRIPTION =
      (ADDRESS = (PROTOCOL = TCP)(HOST = 192.168.71.129)(PORT = 1521))
      (ADDRESS = (PROTOCOL = IPC)(KEY = EXTPROC1521))
    )
  )

XE =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = 192.168.71.129)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = XE)
    )
  )

LISTENER_XE =
  (ADDRESS = (PROTOCOL = TCP)(HOST = 192.168.71.129)(PORT = 1521))

ORACLE_CONNECTION_DATA =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = IPC)(KEY = EXTPROC1521))
    )
    (CONNECT_DATA =
      (SID = CLRExtProc)
      (PRESENTATION = RO)
    )
  )


```

DESPUÉS:

```
# tnsnames.ora Network Configuration File: C:\app\BBM\product\18.0.0\dbhomeXE\NETWORK\ADMIN\tnsnames.ora
# Generated by Oracle configuration tools.

# listener.ora Network Configuration File: C:\app\BBM\product\18.0.0\dbhomeXE\NETWORK\LISTENER.ORA
# Generated by Oracle configuration tools.

DEFAULT_SERVICE_LISTENER = XE

SID_LIST_LISTENER =
  (SID_LIST =
    (SID_DESC =
      (SID_NAME = CLRExtProc)
      (ORACLE_HOME = C:\app\BBM\product\18.0.0\dbhomeXE)
      (PROGRAM = extproc)
      (ENVS = "EXTPROC_DLLS=ONLY:C:\app\BBM\product\18.0.0\dbhomeXE\bin\oraclr18.dll")
    )
  )

LISTENER =
  (DESCRIPTION_LIST =
    (DESCRIPTION =
      (ADDRESS = (PROTOCOL = TCP)(HOST = localhost)(PORT = 1521))
      (ADDRESS = (PROTOCOL = IPC)(KEY = EXTPROC1521))
    )
  )

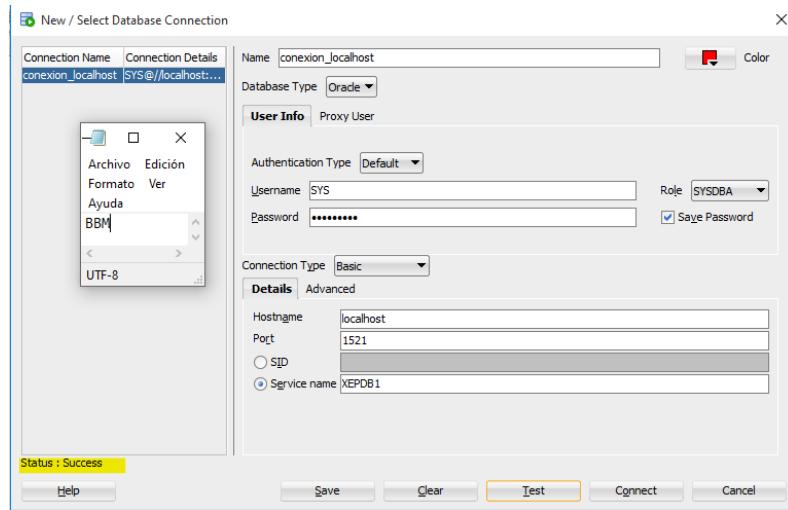
XE =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = localhost)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = XE)
    )
  )

LISTENER_XE =
  (ADDRESS = (PROTOCOL = TCP)(HOST = 192.168.71.129)(PORT = 1521))

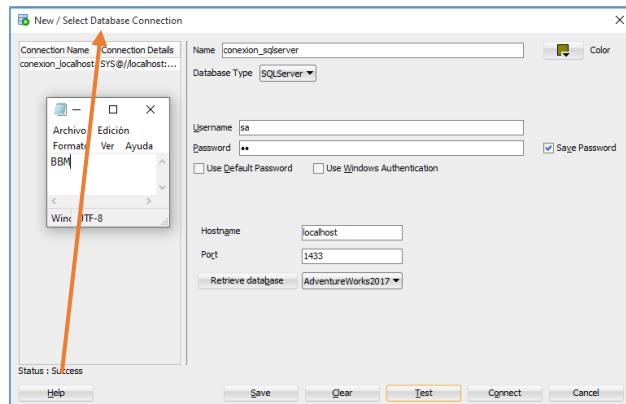
ORACLE_CONNECTION_DATA =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = IPC)(KEY = EXTPROC1521))
    )
    (CONNECT_DATA =
      (SID = CLRExtProc)
      (PRESENTATION = RO)
    )
  )


```

Cuando cambiamos esto, guardamos todos los cambios, reiniciamos todo Oracle y probamos la conexión (en este caso me va a decir que ya la tengo). Si funciona ya nos conectaríamos:

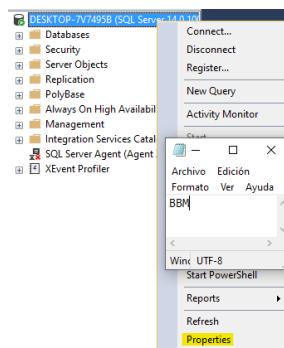


Vamos a demostrar ahora como haríamos la conexión de nuestro SQLDeveloper a SQL Server. La conexión va a tener la estructura de un nombre, un tipo de base de datos, un usuario administrador de SQL Server (**sa**), una contraseña (**sa** en este caso), el Hostname *localhost* de nuevo y el puerto 1433.

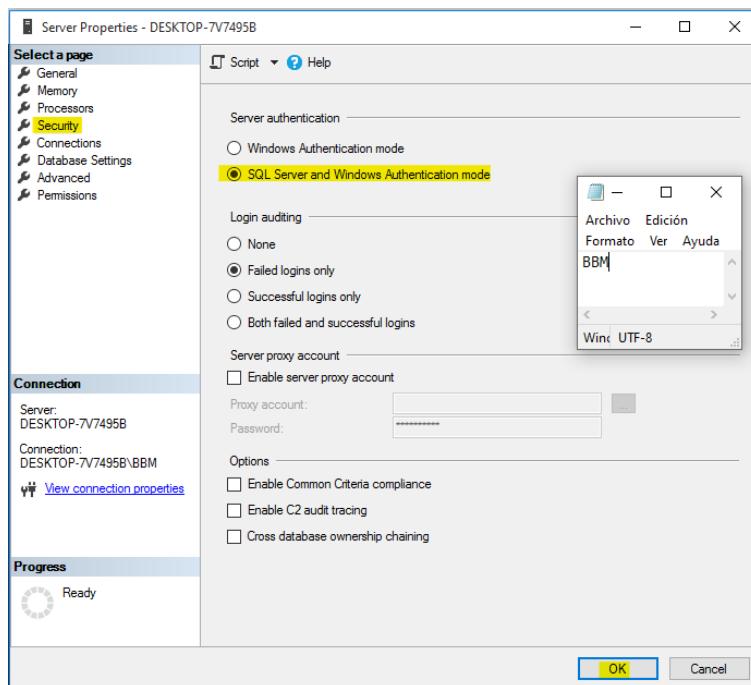


El resultado sería algo como esto:

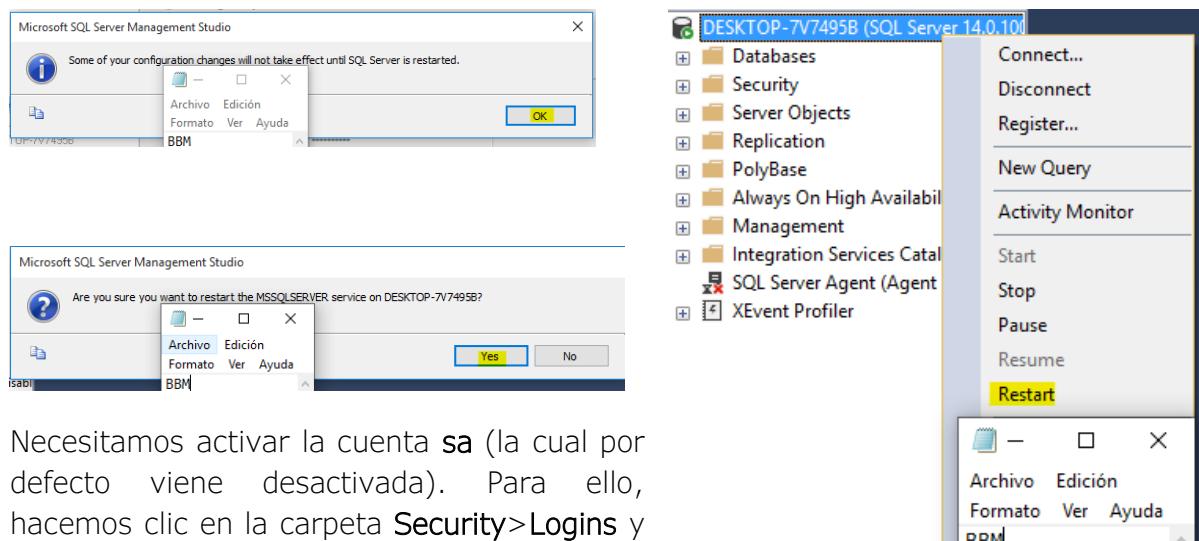
Pero ¿Cómo conseguimos llegar a esto? Mostraré los pasos previos: Primero me voy al SSMS y cambio el **tipo de autenticación**. Para cambiar esto he de ir a la instancia haciendo clic con el botón derecho sobre la misma y pulsar en **Properties**.



En la ventana que aparece le damos a la sección **Security**. Tal y como lo tenemos instalado de antes, en la sección *Server Authentication* tenemos seleccionado *Windows Authentication Mode*, el cual tenemos que cambiar a **SQL Server and Windows Authentication mode** (modo mixto).

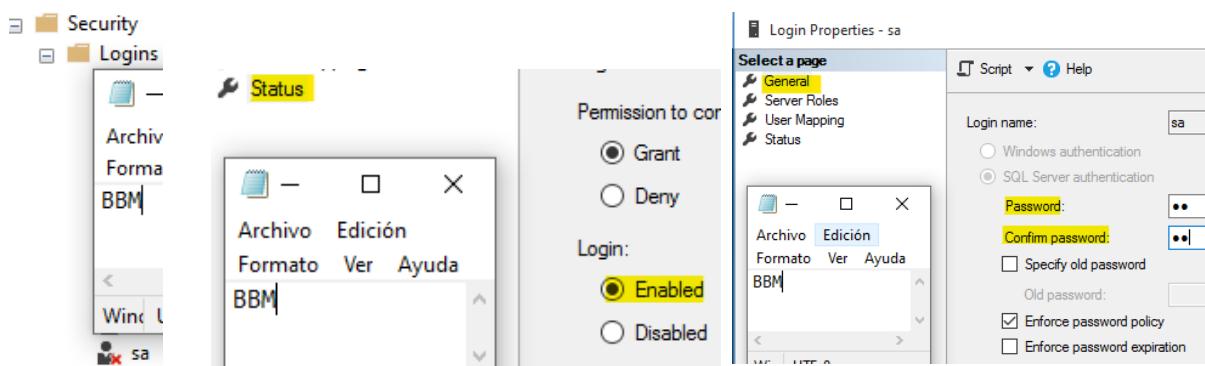


En el momento que le doy a aceptar nos sale una ventana de aviso en el cual nos indica que debemos reiniciar el servidor para que este cambio se haga efectivo. Hacemos clic de botón derecho sobre la instancia y le damos a **Restart**. Al aceptar, el sistema detiene el servicio y lo vuelve a empezar.



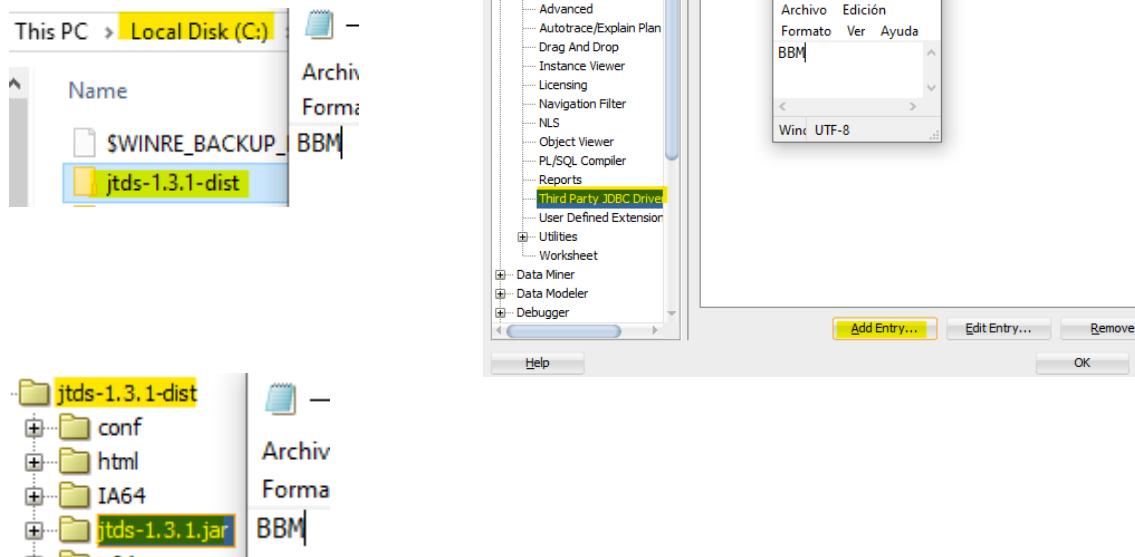
Necesitamos activar la cuenta **sa** (la cual por defecto viene desactivada). Para ello, hacemos clic en la carpeta **Security>Logins** y ahí vemos el icono marcado con una x roja

. Hacemos clic sobre **sa** y hacemos dos cosas: cambiamos la contraseña por **sa** (que no engañe el hecho de que ponga muchos asteriscos/puntos porque por seguridad no sale la longitud de la contraseña que se introduce). En la pestaña **Status** cambiamos el *Login* al estado **Enabled** para que podamos trabajar con él en las conexiones. Sabemos que está activo porque el icono se verá así

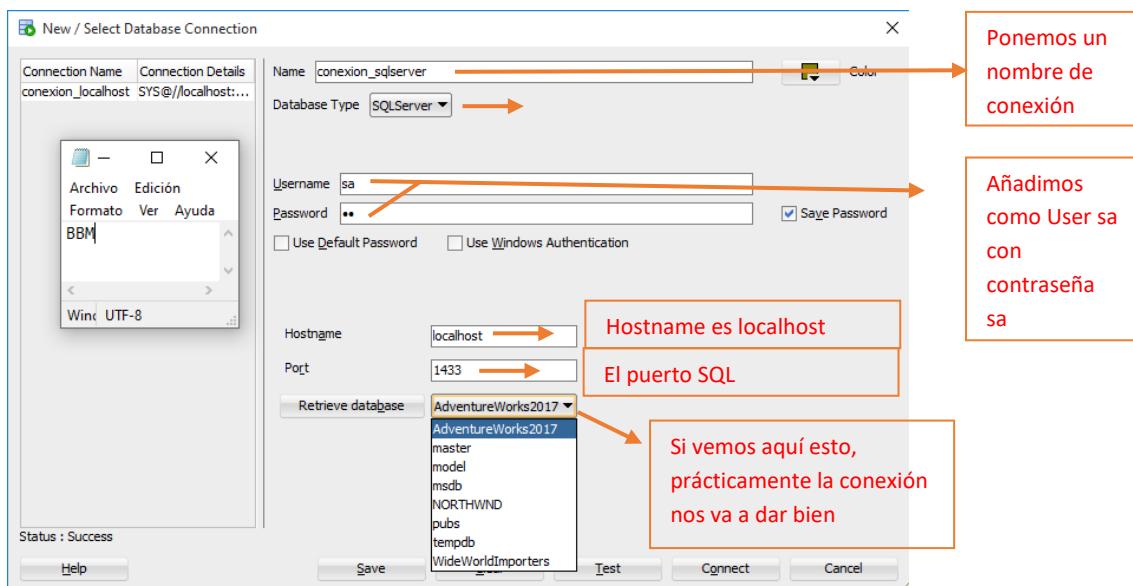


Una vez hecha esta parte, nos vamos a **SQLDeveloper** y comenzamos a configurar la conexión de SQL Server. Si le damos a +, vamos a ver que en el desplegable sólo nos deja la opción de **Oracle**. Para cambiar esto y poder elegir SQL Server, vamos a copiar los drivers de JavaDBC y los copiamos en C:/. Ahora, los pasos que tengo que hacer es ir a **Tools>Preferencias>Pestaña Database>Third Party Driver**. Allí, tendremos que darle a **Add Entry...** para seleccionar (dentro de la carpeta que añadimos a C:/) la

carpeta **jtsd-1.3.1.jar**. Le damos a **Ok** y relanzamos el **SQL Developer**.



Una vez hacemos todos estos pasos, me dispongo a configurar de la siguiente manera la **conexión SQL Server**:



Como se puede ver, ya tengo las distintas bases de datos y los distintos elementos

ORDINAL_POSITION	COLUMN_NAME	DATA_TYPE	IS_NULLABLE	NUMERIC_PRECISION	NUMERIC_SCALE	DATETIME_PRECISION
1	au_id	varchar	NO	(null)		(null)
2	au_lname	varchar	NO	(null)		(null)
3	au_fname	varchar	NO	(null)		(null)
4	phone	char	NO	(null)		(null)
5	address	varchar	YES	(null)		(null)
6	city	varchar	YES	(null)		(null)
7	state	char	YES	(null)		(null)
8	zip	char	YES	(null)		(null)
9	contract	bit	NO	(null)		(null)

 A status bar at the bottom right says 'WinC UTF-8'.

Desbloquear Usuario HR

Una cuenta de usuario Oracle puede estar bloqueada desde el momento de su creación o por algún otro factor, por tanto, para desbloquear una cuenta se debe iniciar sesión con un usuario administrador como **SYS**, **SYSTEM** o cualquier otra cuenta con privilegios **DBA**.

Para desbloquear el usuario hemos de seguir los siguientes pasos:

Hacemos una conexión a la Shell, impersono como SYSDBA.

```
Microsoft Windows [Version 10.0.10240]
(c) 2015 Microsoft Corporation. All rights reserved.
```

```
C:\Windows\system32>sqlplus
```

```
SQL*Plus: Release 18.0.0.0.0 - Production on Tue Nov 10 21:16:54 2020
Version 18.4.0.0.0
```

```
Copyright (c) 1982, 2018, Oracle. All rights reserved.
```

```
Enter user-name: SYS /AS SYSDBA
Enter password:
```

```
Connected to:
Oracle Database 18c Express Edition Release 18.0.0.0.0 - Production
Version 18.4.0.0.0
```

```
SQL> show user;
USER is "SYS"
```

Oracle tiene una base de datos de sistema (**container**) y una base de datos que genera para los usuarios (**pluggable**), que es XEPDB1. Nos situamos en ella:

```
SQL> alter session set container=XEPDB1;
Session altered.
```

Le damos formato a las columnas para que se vean bien distribuidas con el comando **column username format a25;**. Y con el comando **select username, account_status from dba_users;**, veo todos los usuarios de nuestra DBA. Como vemos en el siguiente log, HR está bloqueada.

```
SQL> column username format a25;
SQL> select username, account_status from dba_users;
```

USERNAME	ACCOUNT_STATUS
SYS	OPEN
SYSTEM	OPEN
XS\$NULL	EXPIRED & LOCKED
LBACSYS	LOCKED
OUTLN	EXPIRED & LOCKED
DBSNMP	EXPIRED & LOCKED
APPQOSSYS	EXPIRED & LOCKED
DBSFWUSER	EXPIRED & LOCKED
GGSYS	EXPIRED & LOCKED

ANONYMOUS	EXPIRED & LOCKED
HR	EXPIRED & LOCKED

USERNAME	ACCOUNT_STATUS
CTXSYS	EXPIRED & LOCKED
DVSYS	LOCKED
SI_INFORMTN_SCHEMA	EXPIRED & LOCKED
DVF	LOCKED
GSMADMIN_INTERNAL	EXPIRED & LOCKED
ORDPLUGINS	EXPIRED & LOCKED
MDSYS	EXPIRED & LOCKED
OLAPSYS	EXPIRED & LOCKED
ORDDATA	EXPIRED & LOCKED
XDB	EXPIRED & LOCKED
WMSYS	EXPIRED & LOCKED

USERNAME	ACCOUNT_STATUS
ORDSYS	EXPIRED & LOCKED
GSMCATUSER	EXPIRED & LOCKED
MDDATA	EXPIRED & LOCKED
SYSBACKUP	EXPIRED & LOCKED
REMOTE_SCHEDULER_AGENT	EXPIRED & LOCKED
PDBADMIN	OPEN
GSMUSER	EXPIRED & LOCKED
SYSRAC	EXPIRED & LOCKED
OJVMSYS	LOCKED
AUDSYS	LOCKED
DIP	EXPIRED & LOCKED

USERNAME	ACCOUNT_STATUS
SYSKM	EXPIRED & LOCKED
ORACLE_OCM	EXPIRED & LOCKED
SYSSUMF	EXPIRED & LOCKED
SYSDG	EXPIRED & LOCKED

37 rows selected.

Para desbloquearla usamos el comando **alter user HR identified by HR account unlock;**.

```
SQL> alter user HR identified by HR account unlock;
```

```
User altered.
```

Volvemos a dar el formato a las columnas y comprobamos que el usuario ya existe.

```
SQL> column username format a25;
SQL> select username, account_status from dba_users;
```

USERNAME	ACCOUNT_STATUS
SYS	OPEN
SYSTEM	OPEN
XS\$NULL	EXPIRED & LOCKED

LBACSYS	LOCKED
OUTLN	EXPIRED & LOCKED
DBSNMP	EXPIRED & LOCKED
APPQOSSYS	EXPIRED & LOCKED
DBSFWUSER	EXPIRED & LOCKED
GGSYS	EXPIRED & LOCKED
ANONYMOUS	EXPIRED & LOCKED
HR	OPEN
USERNAME	ACCOUNT_STATUS

CTXSYS	EXPIRED & LOCKED
DVSYS	LOCKED
SI_INFORMTN_SCHEMA	EXPIRED & LOCKED
DVF	LOCKED
GSMADMIN_INTERNAL	EXPIRED & LOCKED
ORDPLUGINS	EXPIRED & LOCKED
MDSYS	EXPIRED & LOCKED
OLAPSYS	EXPIRED & LOCKED
ORDDATA	EXPIRED & LOCKED
XDB	EXPIRED & LOCKED
WMSYS	EXPIRED & LOCKED
USERNAME	ACCOUNT_STATUS

ORDSYS	EXPIRED & LOCKED
GSMCATUSER	EXPIRED & LOCKED
MDDATA	EXPIRED & LOCKED
SYSBACKUP	EXPIRED & LOCKED
REMOTE_SCHEDULER_AGENT	EXPIRED & LOCKED
PDBADMIN	OPEN
GSMUSER	EXPIRED & LOCKED
SYSRAC	EXPIRED & LOCKED
OJVMSYS	LOCKED
AUDSYS	LOCKED
DIP	EXPIRED & LOCKED
USERNAME	ACCOUNT_STATUS

SYSKM	EXPIRED & LOCKED
ORACLE_OCM	EXPIRED & LOCKED
SYS\$UMF	EXPIRED & LOCKED
SYSDG	EXPIRED & LOCKED

37 rows selected.

Probamos que la conexión nos va correctamente en cmd.

```
SQL> exit
Disconnected from Oracle Database 18c Express Edition Release
18.0.0.0 - Production
Version 18.4.0.0.0
```

C:\Windows\system32>sqlplus

```
SQL*Plus: Release 18.0.0.0.0 - Production on Tue Nov 10 21:24:27 2020
Version 18.4.0.0.0
```

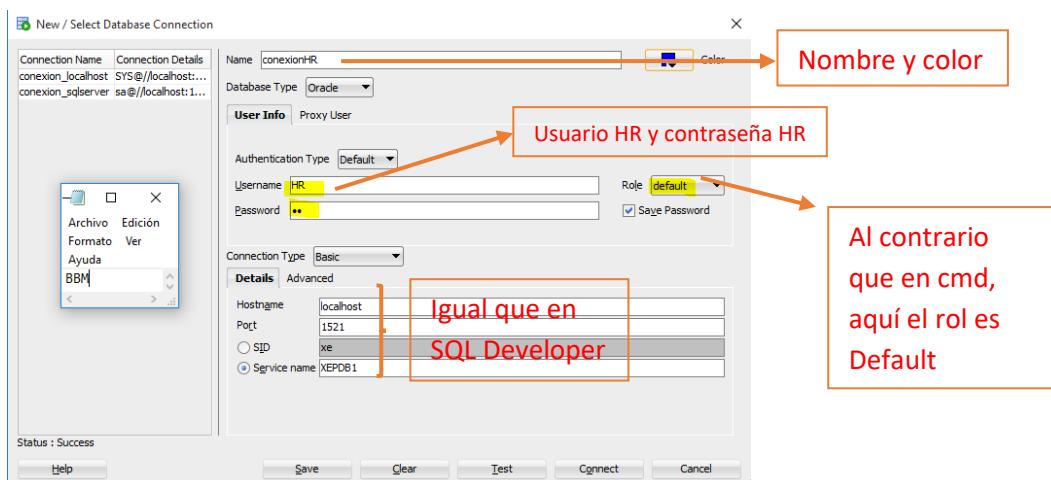
Copyright (c) 1982, 2018, Oracle. All rights reserved.

Enter user-name: HR /AS SYSDBA
Enter password: HR

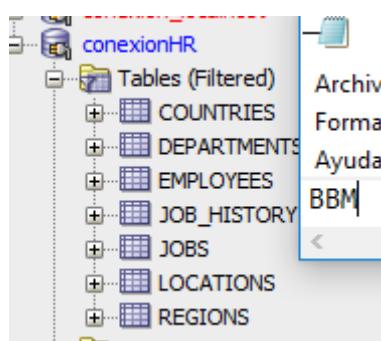
Connected to:
Oracle Database 18c Express Edition Release 18.0.0.0.0 - Production
Version 18.4.0.0.0

NOTA: Como veremos más adelante, en SQL Developer, el rol SYSDBA lo ponemos como Default.

Nos vamos ahora al **SQLDeveloper (GUI)** y creamos la conexión del **usuario HR**. A la hora de hacerlo, clicamos sobre el botón y tenemos que configurarlo de la siguiente manera.



Deberíamos ver esto al acabar la conexión de la siguiente manera



Si pasamos a SQL Server (ya explicaré más adelante cómo) queda un *script* como el siguiente:

```
-- Generated by Oracle SQL Developer Data Modeler 20.2.0.167.1538
-- at:      2020-11-18 17:03:08 CET
-- site:    SQL Server 2008
-- type:   SQL Server 2008
```

```
CREATE TABLE HR.COUNTRIES
(
    COUNTRY_ID CHAR (2) NOT NULL ,
    COUNTRY_NAME VARCHAR (40) ,
    REGION_ID NUMERIC (28)
)
GO

EXEC sp_addextendedproperty 'MS_Description' , 'country table.
Contains 25 rows. References with locations table.' , 'USER' , 'dbo' ,
'TABLE' , 'COUNTRIES'
GO

EXEC sp_addextendedproperty 'MS_Description' , 'Primary key of
countries table.' , 'USER' , 'dbo' , 'TABLE' , 'COUNTRIES' , 'COLUMN'
,'COUNTRY_ID'
GO

EXEC sp_addextendedproperty 'MS_Description' , 'Country name' , 'USER'
,'dbo' , 'TABLE' , 'COUNTRIES' , 'COLUMN' , 'COUNTRY_NAME'
GO

EXEC sp_addextendedproperty 'MS_Description' , 'Region ID for the
country. Foreign key to region_id column in the departments table.' ,
'USER' , 'dbo' , 'TABLE' , 'COUNTRIES' , 'COLUMN' , 'REGION_ID'
GO

ALTER TABLE HR.COUNTRIES ADD CONSTRAINT COUNTRY_C_ID_PK PRIMARY KEY
CLUSTERED (COUNTRY_ID)
    WITH (
        ALLOW_PAGE_LOCKS = ON ,
        ALLOW_ROW_LOCKS = ON )
GO

CREATE TABLE HR.DEPARTMENTS
(
    DEPARTMENT_ID NUMERIC (4) NOT NULL ,
    DEPARTMENT_NAME VARCHAR (30) NOT NULL ,
    MANAGER_ID NUMERIC (6) ,
    LOCATION_ID NUMERIC (4)
)
GO

EXEC sp_addextendedproperty 'MS_Description' , 'Departments table that
shows details of departments where employees
work. Contains 27 rows; references with locations, employees, and
job_history tables.' , 'USER' , 'dbo' , 'TABLE' , 'DEPARTMENTS'
GO
```

```
EXEC sp_addextendedproperty 'MS_Description' , 'Primary key column of
departments table.' , 'USER' , 'dbo' , 'TABLE' , 'DEPARTMENTS' ,
'COLUMN' , 'DEPARTMENT_ID'
GO
```

```
EXEC sp_addextendedproperty 'MS_Description' , 'A not null column that
shows name of a department. Administration,
Marketing, Purchasing, Human Resources, Shipping, IT, Executive,
Public
Relations, Sales, Finance, and Accounting.' , 'USER' , 'dbo' ,
'TABLE' , 'DEPARTMENTS' , 'COLUMN' , 'DEPARTMENT_NAME'
GO
```

```
EXEC sp_addextendedproperty 'MS_Description' , 'Manager_id of a
department. Foreign key to employee_id column of employees table. The
manager_id column of the employee table references this column.' ,
'USER' , 'dbo' , 'TABLE' , 'DEPARTMENTS' , 'COLUMN' , 'MANAGER_ID'
GO
```

```
EXEC sp_addextendedproperty 'MS_Description' , 'Location id where a
department is located. Foreign key to location_id column of locations
table.' , 'USER' , 'dbo' , 'TABLE' , 'DEPARTMENTS' , 'COLUMN' ,
'LOCATION_ID'
GO
```

```
CREATE UNIQUE NONCLUSTERED INDEX
    DEPT_ID_PK ON HR.DEPARTMENTS
    (
        DEPARTMENT_ID
    )
GO
```

```
CREATE NONCLUSTERED INDEX
    DEPT_LOCATION_IX ON HR.DEPARTMENTS
    (
        LOCATION_ID
    )
GO
```

```
ALTER TABLE HR.DEPARTMENTS ADD CONSTRAINT DEPT_ID_PK PRIMARY KEY
CLUSTERED (DEPARTMENT_ID)
    WITH (
        ALLOW_PAGE_LOCKS = ON ,
        ALLOW_ROW_LOCKS = ON )
GO
```

```
CREATE TABLE HR.EMPLOYEES
(
    EMPLOYEE_ID NUMERIC (6) NOT NULL ,
```

```
FIRST_NAME VARCHAR (20) ,
LAST_NAME VARCHAR (25) NOT NULL ,
EMAIL VARCHAR (25) NOT NULL ,
PHONE_NUMBER VARCHAR (20) ,
HIRE_DATE DATE NOT NULL ,
JOB_ID VARCHAR (10) NOT NULL ,
SALARY NUMERIC (8,2) ,
COMMISSION_PCT NUMERIC (2,2) ,
MANAGER_ID NUMERIC (6) ,
DEPARTMENT_ID NUMERIC (4)
)
GO

ALTER TABLE HR.EMPLOYEES
ADD CONSTRAINT EMP_SALARY_MIN
CHECK ( salary > 0 )
GO

EXEC sp_addextendedproperty 'MS_Description' , 'employees table.
Contains 107 rows. References with departments,
jobs, job_history tables. Contains a self reference.' , 'USER' , 'dbo'
, 'TABLE' , 'EMPLOYEES'
GO

EXEC sp_addextendedproperty 'MS_Description' , 'Primary key of
employees table.' , 'USER' , 'dbo' , 'TABLE' , 'EMPLOYEES' , 'COLUMN'
, 'EMPLOYEE_ID'
GO

EXEC sp_addextendedproperty 'MS_Description' , 'First name of the
employee. A not null column.' , 'USER' , 'dbo' , 'TABLE' , 'EMPLOYEES'
, 'COLUMN' , 'FIRST_NAME'
GO

EXEC sp_addextendedproperty 'MS_Description' , 'Last name of the
employee. A not null column.' , 'USER' , 'dbo' , 'TABLE' , 'EMPLOYEES'
, 'COLUMN' , 'LAST_NAME'
GO

EXEC sp_addextendedproperty 'MS_Description' , 'Email id of the
employee' , 'USER' , 'dbo' , 'TABLE' , 'EMPLOYEES' , 'COLUMN' ,
'EMAIL'
GO

EXEC sp_addextendedproperty 'MS_Description' , 'Phone number of the
employee; includes country code and area code' , 'USER' , 'dbo' ,
'TABLE' , 'EMPLOYEES' , 'COLUMN' , 'PHONE_NUMBER'
GO
```

```
EXEC sp_addextendedproperty 'MS_Description' , 'Date when the employee started on this job. A not null column.' , 'USER' , 'dbo' , 'TABLE' , 'EMPLOYEES' , 'COLUMN' , 'HIRE_DATE'  
GO
```

```
EXEC sp_addextendedproperty 'MS_Description' , 'Current job of the employee; foreign key to job_id column of the jobs table. A not null column.' , 'USER' , 'dbo' , 'TABLE' , 'EMPLOYEES' , 'COLUMN' , 'JOB_ID'  
GO
```

```
EXEC sp_addextendedproperty 'MS_Description' , 'Monthly salary of the employee. Must be greater than zero (enforced by constraint emp_salary_min)' , 'USER' , 'dbo' , 'TABLE' , 'EMPLOYEES' , 'COLUMN' , 'SALARY'  
GO
```

```
EXEC sp_addextendedproperty 'MS_Description' , 'Commission percentage of the employee; Only employees in sales department eligible for commission percentage' , 'USER' , 'dbo' , 'TABLE' , 'EMPLOYEES' , 'COLUMN' , 'COMMISSION_PCT'  
GO
```

```
EXEC sp_addextendedproperty 'MS_Description' , 'Manager id of the employee; has same domain as manager_id in departments table. Foreign key to employee_id column of employees table.  
(useful for reflexive joins and CONNECT BY query)' , 'USER' , 'dbo' , 'TABLE' , 'EMPLOYEES' , 'COLUMN' , 'MANAGER_ID'  
GO
```

```
EXEC sp_addextendedproperty 'MS_Description' , 'Department id where employee works; foreign key to department_id column of the departments table' , 'USER' , 'dbo' , 'TABLE' , 'EMPLOYEES' , 'COLUMN' , 'DEPARTMENT_ID'  
GO
```

```
CREATE NONCLUSTERED INDEX  
EMP_DEPARTMENT_IX ON HR.EMPLOYEES  
(  
    DEPARTMENT_ID  
)  
GO
```

```
CREATE UNIQUE NONCLUSTERED INDEX  
    EMP_EMAIL_UK ON HR.EMPLOYEES  
    (  
        EMAIL  
    )  
GO
```

```
CREATE UNIQUE NONCLUSTERED INDEX  
    EMP_EMP_ID_PK ON HR.EMPLOYEES  
    (  
        EMPLOYEE_ID  
    )  
GO
```

```
CREATE NONCLUSTERED INDEX  
    EMP_JOB_IK ON HR.EMPLOYEES  
    (  
        JOB_ID  
    )  
GO
```

```
CREATE NONCLUSTERED INDEX  
    EMP_MANAGER_IK ON HR.EMPLOYEES  
    (  
        MANAGER_ID  
    )  
GO
```

```
CREATE NONCLUSTERED INDEX  
    EMP_NAME_IK ON HR.EMPLOYEES  
    (  
        LAST_NAME ,  
        FIRST_NAME  
    )  
GO
```

```
ALTER TABLE HR.EMPLOYEES ADD CONSTRAINT EMP_EMP_ID_PK PRIMARY KEY  
CLUSTERED (EMPLOYEE_ID)  
    WITH (  
        ALLOW_PAGE_LOCKS = ON ,  
        ALLOW_ROW_LOCKS = ON )  
GO  
ALTER TABLE HR.EMPLOYEES ADD CONSTRAINT EMP_EMAIL_UK UNIQUE  
NONCLUSTERED (EMAIL)  
GO
```

```
CREATE TABLE HR.JOB_HISTORY  
    (  
        EMPLOYEE_ID NUMERIC (6) NOT NULL ,  
        START_DATE DATE NOT NULL ,  
        END_DATE DATE NOT NULL ,  
        JOB_ID VARCHAR (10) NOT NULL ,  
        DEPARTMENT_ID NUMERIC (4)  
    )  
GO
```

```
EXEC sp_addextendedproperty 'MS_Description' , 'Table that stores job history of the employees. If an employee changes departments within the job or changes jobs within the department, new rows get inserted into this table with old job information of the employee. Contains a complex primary key: employee_id+start_date. Contains 25 rows. References with jobs, employees, and departments tables.' , 'USER' , 'dbo' , 'TABLE' , 'JOB_HISTORY'  
GO
```

```
EXEC sp_addextendedproperty 'MS_Description' , 'A not null column in the complex primary key employee_id+start_date.  
Foreign key to employee_id column of the employee table' , 'USER' , 'dbo' , 'TABLE' , 'JOB_HISTORY' , 'COLUMN' , 'EMPLOYEE_ID'  
GO
```

```
EXEC sp_addextendedproperty 'MS_Description' , 'A not null column in the complex primary key employee_id+start_date.  
Must be less than the end_date of the job_history table. (enforced by constraint jhist_date_interval)' , 'USER' , 'dbo' , 'TABLE' , 'JOB_HISTORY' , 'COLUMN' , 'START_DATE'  
GO
```

```
EXEC sp_addextendedproperty 'MS_Description' , 'Last day of the employee in this job role. A not null column. Must be greater than the start_date of the job_history table.  
(enforced by constraint jhist_date_interval)' , 'USER' , 'dbo' , 'TABLE' , 'JOB_HISTORY' , 'COLUMN' , 'END_DATE'  
GO
```

```
EXEC sp_addextendedproperty 'MS_Description' , 'Job role in which the employee worked in the past; foreign key to job_id column in the jobs table. A not null column.' , 'USER' , 'dbo' , 'TABLE' , 'JOB_HISTORY' , 'COLUMN' , 'JOB_ID'  
GO
```

```
EXEC sp_addextendedproperty 'MS_Description' , 'Department id in which the employee worked in the past; foreign key to department_id column in the departments table' , 'USER' , 'dbo' , 'TABLE' , 'JOB_HISTORY' , 'COLUMN' , 'DEPARTMENT_ID'  
GO
```

```
CREATE NONCLUSTERED INDEX JHIST_DEPARTMENT_IX ON HR.JOB_HISTORY  
(  
    DEPARTMENT_ID  
)
```

```
GO
```

```
CREATE NONCLUSTERED INDEX
    JHIST_EMPLOYEE_IX ON HR.JOB_HISTORY
    (
        EMPLOYEE_ID
    )
GO
```

```
CREATE UNIQUE NONCLUSTERED INDEX
    JHIST_EMP_ID_ST_DATE_PK ON HR.JOB_HISTORY
    (
        EMPLOYEE_ID ,
        START_DATE
    )
GO
```

```
CREATE NONCLUSTERED INDEX
    JHIST_JOB_IX ON HR.JOB_HISTORY
    (
        JOB_ID
    )
GO
```

```
ALTER TABLE HR.JOB_HISTORY ADD CONSTRAINT JHIST_EMP_ID_ST_DATE_PK
PRIMARY KEY CLUSTERED (EMPLOYEE_ID, START_DATE)
    WITH (
        ALLOW_PAGE_LOCKS = ON ,
        ALLOW_ROW_LOCKS = ON )
GO
```

```
ALTER TABLE HR.JOB_HISTORY ADD CONSTRAINT JHIST_DATE_INTERVAL CHECK (
end_date > start_date )
GO
```

```
CREATE TABLE HR.JOBS
(
    JOB_ID VARCHAR (10) NOT NULL ,
    JOB_TITLE VARCHAR (35) NOT NULL ,
    MIN_SALARY NUMERIC (6) ,
    MAX_SALARY NUMERIC (6)
)
GO
```

```
EXEC sp_addextendedproperty 'MS_Description' , 'jobs table with job
titles and salary ranges. Contains 19 rows.
References with employees and job_history table.' , 'USER' , 'dbo' ,
'TABLE' , 'JOBS'
GO
```

```
EXEC sp_addextendedproperty 'MS_Description' , 'Primary key of jobs
table.' , 'USER' , 'dbo' , 'TABLE' , 'JOBS' , 'COLUMN' , 'JOB_ID'
GO
```

```
EXEC sp_addextendedproperty 'MS_Description' , 'A not null column that
shows job title, e.g. AD_VP, FI_ACCOUNTANT' , 'USER' , 'dbo' , 'TABLE'
, 'JOBS' , 'COLUMN' , 'JOB_TITLE'
GO
```

```
EXEC sp_addextendedproperty 'MS_Description' , 'Minimum salary for a
job title.' , 'USER' , 'dbo' , 'TABLE' , 'JOBS' , 'COLUMN'
, 'MIN_SALARY'
GO
```

```
EXEC sp_addextendedproperty 'MS_Description' , 'Maximum salary for a
job title' , 'USER' , 'dbo' , 'TABLE' , 'JOBS' , 'COLUMN'
, 'MAX_SALARY'
GO
```

```
CREATE UNIQUE NONCLUSTERED INDEX
    JOB_ID_PK ON HR.JOBS
    (
        JOB_ID
    )
GO

ALTER TABLE HR.JOB ADD CONSTRAINT JOB_ID_PK PRIMARY KEY CLUSTERED
(JOB_ID)
    WITH (
        ALLOW_PAGE_LOCKS = ON ,
        ALLOW_ROW_LOCKS = ON )
GO
```

```
CREATE TABLE HR.LOCATIONS
(
    LOCATION_ID NUMERIC (4) NOT NULL ,
    STREET_ADDRESS VARCHAR (40) ,
    POSTAL_CODE VARCHAR (12) ,
    CITY VARCHAR (30) NOT NULL ,
    STATE_PROVINCE VARCHAR (25) ,
    COUNTRY_ID CHAR (2)
)
GO
```

```
EXEC sp_addextendedproperty 'MS_Description' , 'Locations table that
contains specific address of a specific office,
warehouse, and/or production site of a company. Does not store
addresses /
locations of customers. Contains 23 rows; references with the
departments and countries tables.' , 'USER' , 'dbo' , 'TABLE'
, 'LOCATIONS'
GO
```

```
EXEC sp_addextendedproperty 'MS_Description' , 'Primary key of locations table' , 'USER' , 'dbo' , 'TABLE' , 'LOCATIONS' , 'COLUMN' , 'LOCATION_ID'
GO
```

```
EXEC sp_addextendedproperty 'MS_Description' , 'Street address of an office, warehouse, or production site of a company. Contains building number and street name' , 'USER' , 'dbo' , 'TABLE' , 'LOCATIONS' , 'COLUMN' , 'STREET_ADDRESS'
GO
```

```
EXEC sp_addextendedproperty 'MS_Description' , 'Postal code of the location of an office, warehouse, or production site of a company.' , 'USER' , 'dbo' , 'TABLE' , 'LOCATIONS' , 'COLUMN' , 'POSTAL_CODE'
GO
```

```
EXEC sp_addextendedproperty 'MS_Description' , 'A not null column that shows city where an office, warehouse, or production site of a company is located.' , 'USER' , 'dbo' , 'TABLE' , 'LOCATIONS' , 'COLUMN' , 'CITY'
GO
```

```
EXEC sp_addextendedproperty 'MS_Description' , 'State or Province where an office, warehouse, or production site of a company is located.' , 'USER' , 'dbo' , 'TABLE' , 'LOCATIONS' , 'COLUMN' , 'STATE_PROVINCE'
GO
```

```
EXEC sp_addextendedproperty 'MS_Description' , 'Country where an office, warehouse, or production site of a company is located. Foreign key to country_id column of the countries table.' , 'USER' , 'dbo' , 'TABLE' , 'LOCATIONS' , 'COLUMN' , 'COUNTRY_ID'
GO
```

```
CREATE NONCLUSTERED INDEX LOC_CITY_IX ON HR.LOCATIONS
(
    CITY
)
GO
```

```
CREATE NONCLUSTERED INDEX LOC_COUNTRY_IX ON HR.LOCATIONS
(
    COUNTRY_ID
)
```

```
)  
GO  
  
CREATE UNIQUE NONCLUSTERED INDEX  
LOC_ID_PK ON HR.LOCATIONS  
(  
LOCATION_ID  
)  
GO  
  
CREATE NONCLUSTERED INDEX  
LOC_STATE_PROVINCE_IX ON HR.LOCATIONS  
(  
STATE_PROVINCE  
)  
GO  
  
ALTER TABLE HR.LOCATIONS ADD CONSTRAINT LOC_ID_PK PRIMARY KEY  
CLUSTERED (LOCATION_ID)  
WITH (  
ALLOW_PAGE_LOCKS = ON ,  
ALLOW_ROW_LOCKS = ON )  
GO  
  
CREATE TABLE HR.REGIONS  
(  
REGION_ID NUMERIC (28) NOT NULL ,  
REGION_NAME VARCHAR (25)  
)  
GO  
  
CREATE UNIQUE NONCLUSTERED INDEX  
REG_ID_PK ON HR.REGIONS  
(  
REGION_ID  
)  
GO  
  
ALTER TABLE HR.REGIONS ADD CONSTRAINT REG_ID_PK PRIMARY KEY CLUSTERED  
(REGION_ID)  
WITH (  
ALLOW_PAGE_LOCKS = ON ,  
ALLOW_ROW_LOCKS = ON )  
GO  
  
ALTER TABLE HR.COUNTRIES  
ADD CONSTRAINT COUNTR_REG_FK FOREIGN KEY  
(  
REGION_ID  
)  
REFERENCES HR.REGIONS  
(  
REGION_ID  
)  
ON DELETE NO ACTION  
ON UPDATE NO ACTION
```

GO

```
ALTER TABLE HR.DEPARTMENTS
  ADD CONSTRAINT DEPT_LOC_FK FOREIGN KEY
  (
    LOCATION_ID
  )
  REFERENCES HR.LOCATIONS
  (
    LOCATION_ID
  )
  ON DELETE NO ACTION
  ON UPDATE NO ACTION
GO
```

```
ALTER TABLE HR.DEPARTMENTS
  ADD CONSTRAINT DEPT_MGR_FK FOREIGN KEY
  (
    MANAGER_ID
  )
  REFERENCES HR.EMPLOYEES
  (
    EMPLOYEE_ID
  )
  ON DELETE NO ACTION
  ON UPDATE NO ACTION
GO
```

```
ALTER TABLE HR.EMPLOYEES
  ADD CONSTRAINT EMP_DEPT_FK FOREIGN KEY
  (
    DEPARTMENT_ID
  )
  REFERENCES HR.DEPARTMENTS
  (
    DEPARTMENT_ID
  )
  ON DELETE NO ACTION
  ON UPDATE NO ACTION
GO
```

```
ALTER TABLE HR.EMPLOYEES
  ADD CONSTRAINT EMP_JOB_FK FOREIGN KEY
  (
    JOB_ID
  )
  REFERENCES HR.JOBS
  (
    JOB_ID
  )
  ON DELETE NO ACTION
  ON UPDATE NO ACTION
GO
```

```
ALTER TABLE HR.EMPLOYEES
  ADD CONSTRAINT EMP_MANAGER_FK FOREIGN KEY
  (
    MANAGER_ID
  )
  REFERENCES HR.EMPLOYEES
  (
```

```
EMPLOYEE_ID
)
ON DELETE NO ACTION
ON UPDATE NO ACTION
GO

ALTER TABLE HR.JOB_HISTORY
ADD CONSTRAINT JHIST_DEPT_FK FOREIGN KEY
(
  DEPARTMENT_ID
)
REFERENCES HR.DEPARTMENTS
(
  DEPARTMENT_ID
)
ON DELETE NO ACTION
ON UPDATE NO ACTION
GO

ALTER TABLE HR.JOB_HISTORY
ADD CONSTRAINT JHIST_EMP_FK FOREIGN KEY
(
  EMPLOYEE_ID
)
REFERENCES HR.EMPLOYEES
(
  EMPLOYEE_ID
)
ON DELETE NO ACTION
ON UPDATE NO ACTION
GO

ALTER TABLE HR.JOB_HISTORY
ADD CONSTRAINT JHIST_JOB_FK FOREIGN KEY
(
  JOB_ID
)
REFERENCES HR.JOBS
(
  JOB_ID
)
ON DELETE NO ACTION
ON UPDATE NO ACTION
GO

ALTER TABLE HR.LOCATIONS
ADD CONSTRAINT LOC_C_ID_FK FOREIGN KEY
(
  COUNTRY_ID
)
REFERENCES HR.COUNTRIES
(
  COUNTRY_ID
)
ON DELETE NO ACTION
ON UPDATE NO ACTION
GO
```

-- Oracle SQL Developer Data Modeler Summary Report:

```
--                                         7
-- CREATE TABLE                         18
-- CREATE INDEX                          20
-- ALTER TABLE                           0
-- CREATE VIEW                            0
-- ALTER VIEW                            0
-- CREATE PACKAGE                         0
-- CREATE PACKAGE BODY                   0
-- CREATE PROCEDURE                       0
-- CREATE FUNCTION                        0
-- CREATE TRIGGER                         0
-- ALTER TRIGGER                          0
-- CREATE DATABASE                        0
-- CREATE DEFAULT                         0
-- CREATE INDEX ON VIEW                  0
-- CREATE ROLLBACK SEGMENT                0
-- CREATE ROLE                            0
-- CREATE RULE                            0
-- CREATE SCHEMA                           0
-- CREATE PARTITION FUNCTION             0
-- CREATE PARTITION SCHEME               0
--
-- DROP DATABASE                         0
--
-- ERRORS                                0
-- WARNINGS                               0
```

Bases de Datos de Ejemplo

Vamos a crear un usuario con sus roles, espacio de tabla y conexión.

Los requisitos para hacer esta prueba son los siguientes

- Tener los servicios OracleServiceXE y OracleOraDB18Home1TNSListener iniciados.

Nombre	Estado	Configuración
OracleJobSchedulerXE	Disabled	NT SERVIC...
OracleOraDB18Home1MTSRecov...	Running	Automatic
OracleOraDB18Home1TNSListener	Running	Automatic
OracleServiceXE	Running	Automatic

- Tener instalado *Oracle 18c Express Edition*.

Entramos en el *cmd* como administradores (ejecutar como Administrador) e iniciaremos SQL*Plus entrando con el rol sys /as sysdba.

Vemos la base de datos a la que nos podemos conectar y cambiamos a la sesión de nuestra base de datos predeterminada con el comando `ALTER SESSION SET container = XEPDB1;`

```
Microsoft Windows [Version 10.0.18362.30]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\WINDOWS\system32>sqlplus

SQL*Plus: Release 18.0.0.0.0 - Production on Fri Nov 13 15:11:02 2020
Version 18.4.0.0.0

Copyright (c) 1982, 2018, Oracle. All rights reserved.

Enter user-name: SYS /AS SYSDBA
Enter password:

Connected to:
Oracle Database 18c Express Edition Release 18.0.0.0.0 - Production
Version 18.4.0.0.0

SQL> SELECT name, con_id FROM v$pdbs;

NAME
-----
CON_ID
-----
PDB$SEED
      2

XEPDB1
      3

SQL> ALTER SESSION SET container = XEPDB1;
Session altered
```

Ahora comprobamos si nuestra base de datos predeterminada está en estado **abierta**. En este caso lo hacemos de dos formas, una con el comando en sí para comprobarlo, y otro abriendo la base de datos (lo que nos dará fallo)

```
SQL> SELECT name, con_id FROM v$pdbs;
```

NAME	CON_ID
XEPDB1	3

```
SQL> ALTER PLUGGABLE DATABASE open;
ALTER PLUGGABLE DATABASE open
*
ERROR at line 1:
ORA-65019: pluggable database XEPDB1 already open
```

Creamos el **espacio de tabla** (tablespace → es una unidad de almacenamiento de base de datos que equivale aproximadamente a un grupo de archivos de Microsoft SQL Server. Los espacios de tabla permiten el almacenamiento y la administración de objetos de base de datos en grupos individuales) de nombre **SAMPLETAB**.

```
SQL> CREATE SMALLFILE TABLESPACE SAMPLETAB DATAFILE
'C:\app\BBM\product\18.0.0\oradata\XE\XEPDB1\SAMPLETAB.DBF' SIZE 200M
LOGGING EXTENT MANAGEMENT LOCAL SEGMENT SPACE MANAGEMENT AUTO;
Tablespace created.
```

Creamos nuestro usuario de nombre: BBM, perfil por defecto: DEFAULT, le ponemos de contraseña: oracle, asignamos tablespace por defecto: SAMPLETAB, tablespace temporal: TEMP, desbloqueamos el usuario y le damos una quota ilimitada a nuestro tablespace. Una vez hecho esto, es donde **podemos crear nuestra base de datos**.

```
SQL> CREATE USER BBM PROFILE DEFAULT IDENTIFIED BY oracle DEFAULT
TABLESPACE SAMPLETAB TEMPORARY TABLESPACE TEMP ACCOUNT UNLOCK quota
unlimited on SAMPLETAB;
User created.
```

Antes de poder hacer nada, realmente lo que hacemos es como *SYSDBA* damos uso permisos al usuario que creamos (BBM en este caso), nos impersonamos en el comando **ALTER SESSION SET CURRENT_SCHEMA = BBM**; y por último nos creamos una conexión la cual probaré también en el entorno gráfico. Los datos son los siguientes:

- Contraseña oracle
- Host Name localhost
- Port 1521
- Service Name XEPDB1

Conexión *CMD*

```
SQL> GRANT CONNECT TO BBM;
```

```
Grant succeeded.

SQL> GRANT RESOURCE TO BBM;

Grant succeeded.

SQL> GRANT INSERT ANY TABLE TO BBM;

Grant succeeded.

SQL> ALTER SESSION SET CURRENT_SCHEMA = BBM;

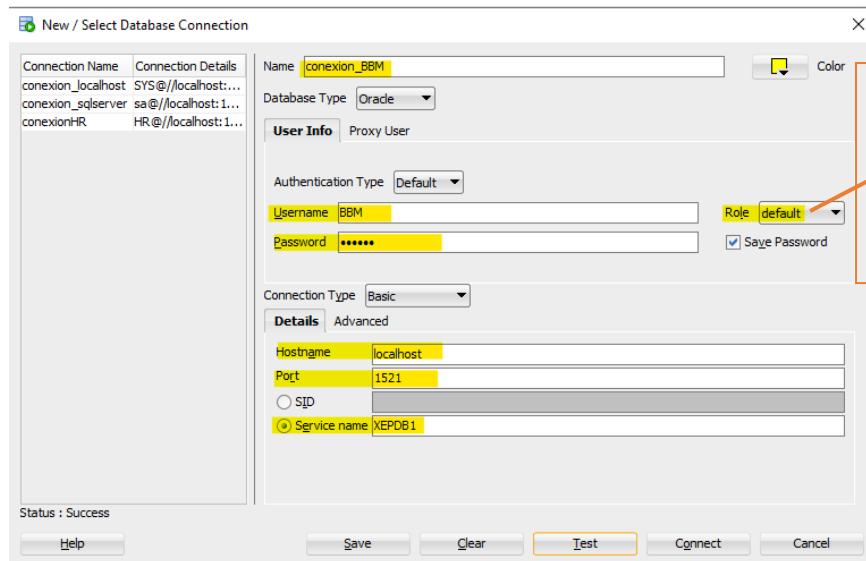
Session altered.

SQL> conn BBM/oracle@//localhost:1521/xepdb1
Connected.
SQL> show user;
USER is "BBM"
SQL> select table_name from user_tables;

no rows selected

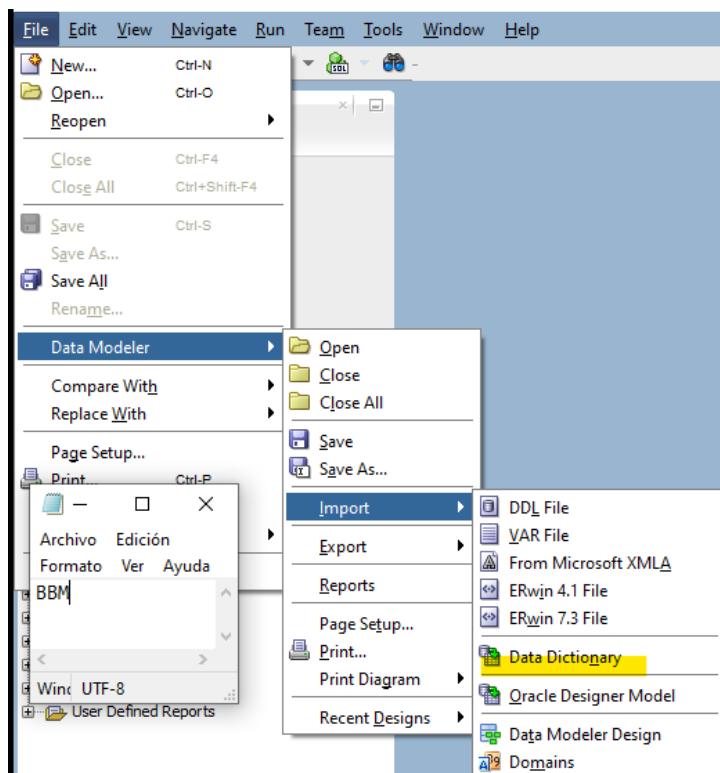
SQL> SHOW USER;
USER is "BBM"
```

Conexión GUI

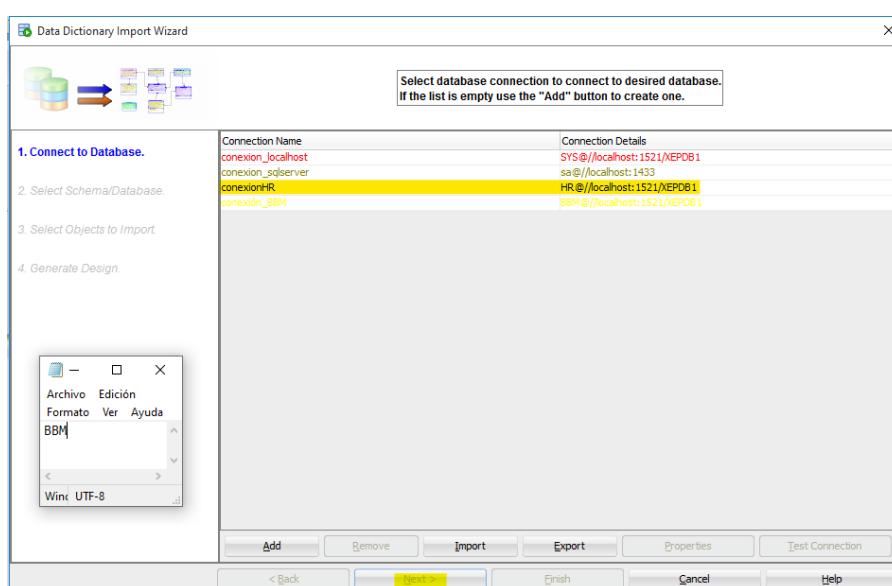


Para hacer una prueba de modelado (cosas que veremos más adelante), haremos una prueba tanto en *SQL Developer* como en el **Data Modeler**. La prueba consistirá en hacer una **ingeniería inversa de modelo relacional a modelo lógico**. Una vez importado el usuario HR y generado el diseño (Punto 2.1.9). Aquí tendríamos el **modelo relacional importado**:

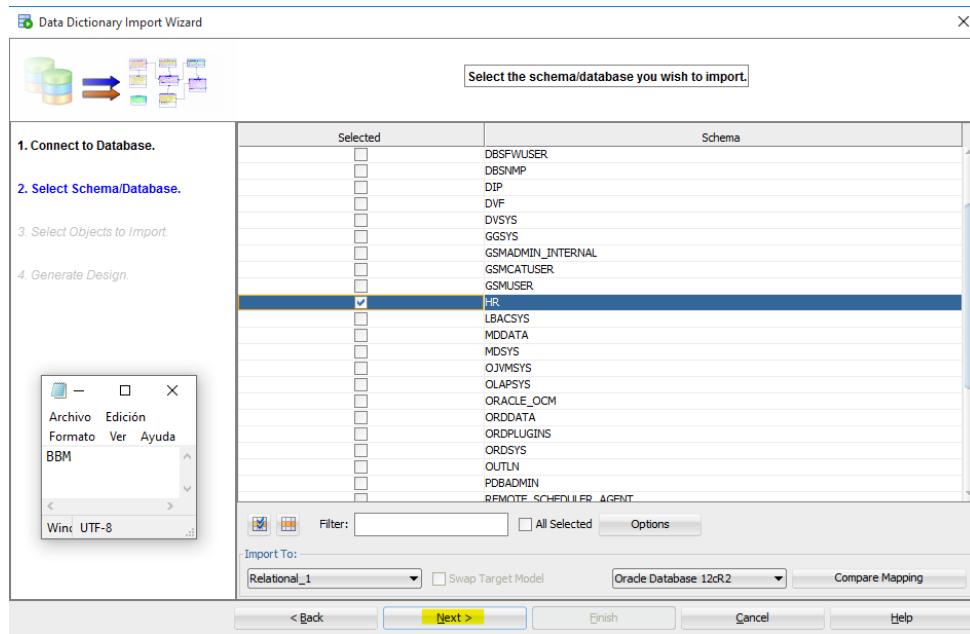
Primero explicaré cómo hacer la ingeniería con *SQL Developer* en GUI. Para ello, hemos de acceder al **Data Modeler** desde *SQL Developer*, hago clic en la siguiente ruta **File>Data Modeler>Import>Data dictionary**.



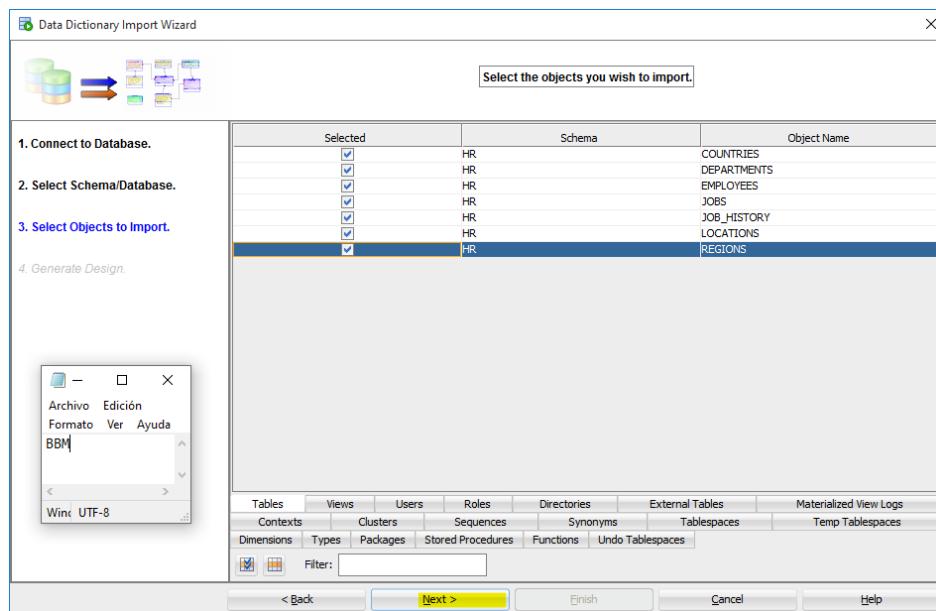
Van a salir las conexiones. Elegimos la que quiero (**HR** en este caso) y le doy a **Next**.



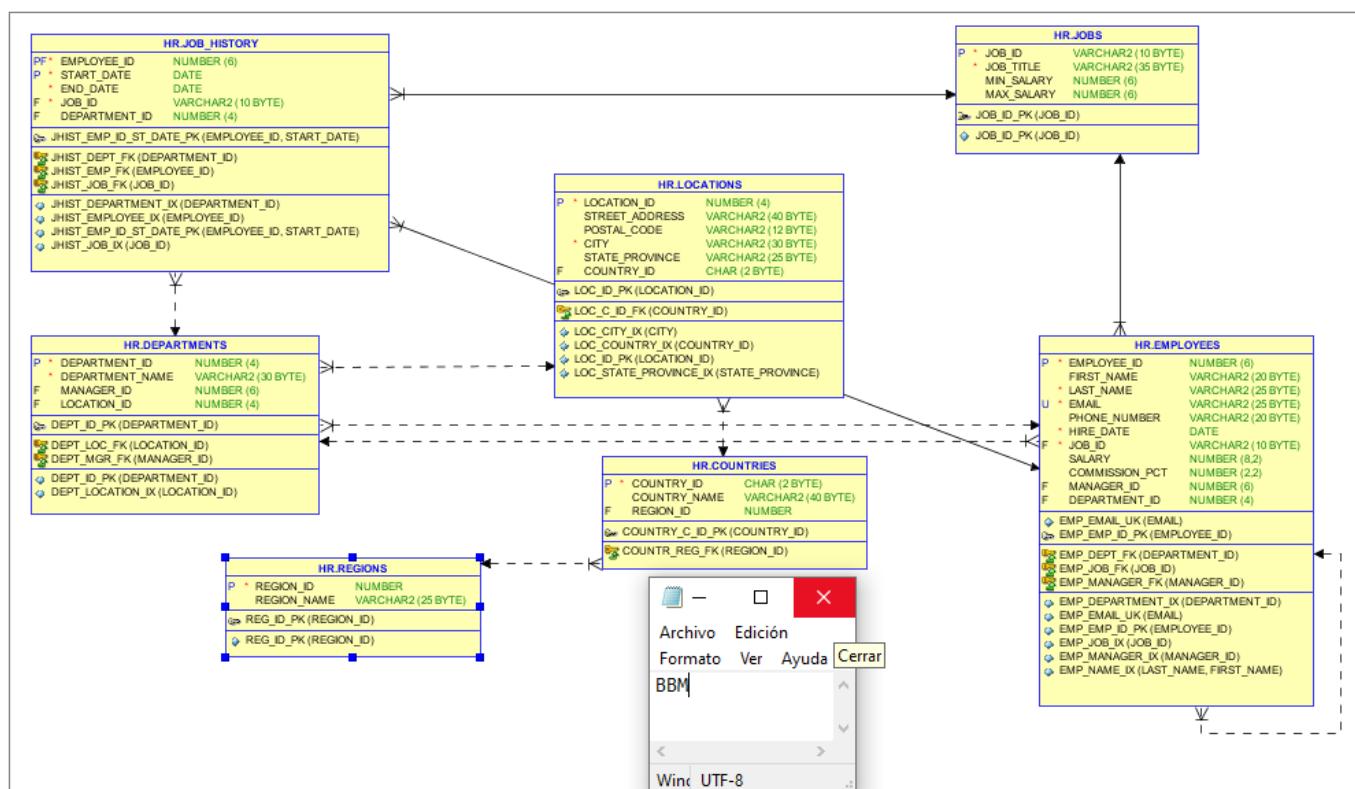
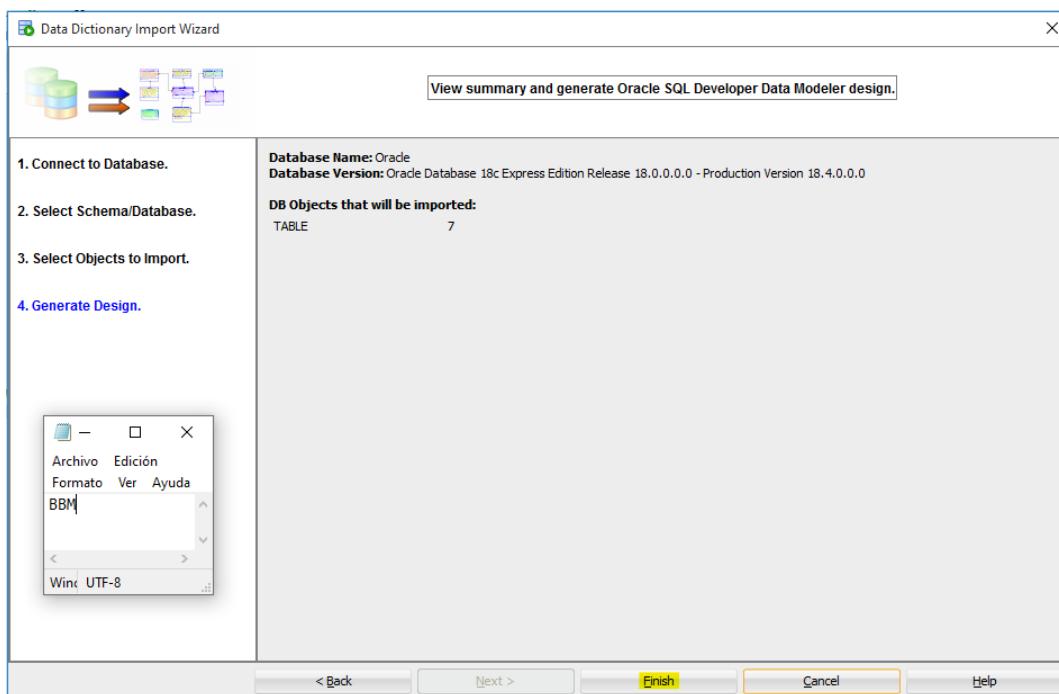
La siguiente ventana son los esquemas (en este caso me interesa elegir HR). Le volvemos a dar *Next* y aparecen las tablas.



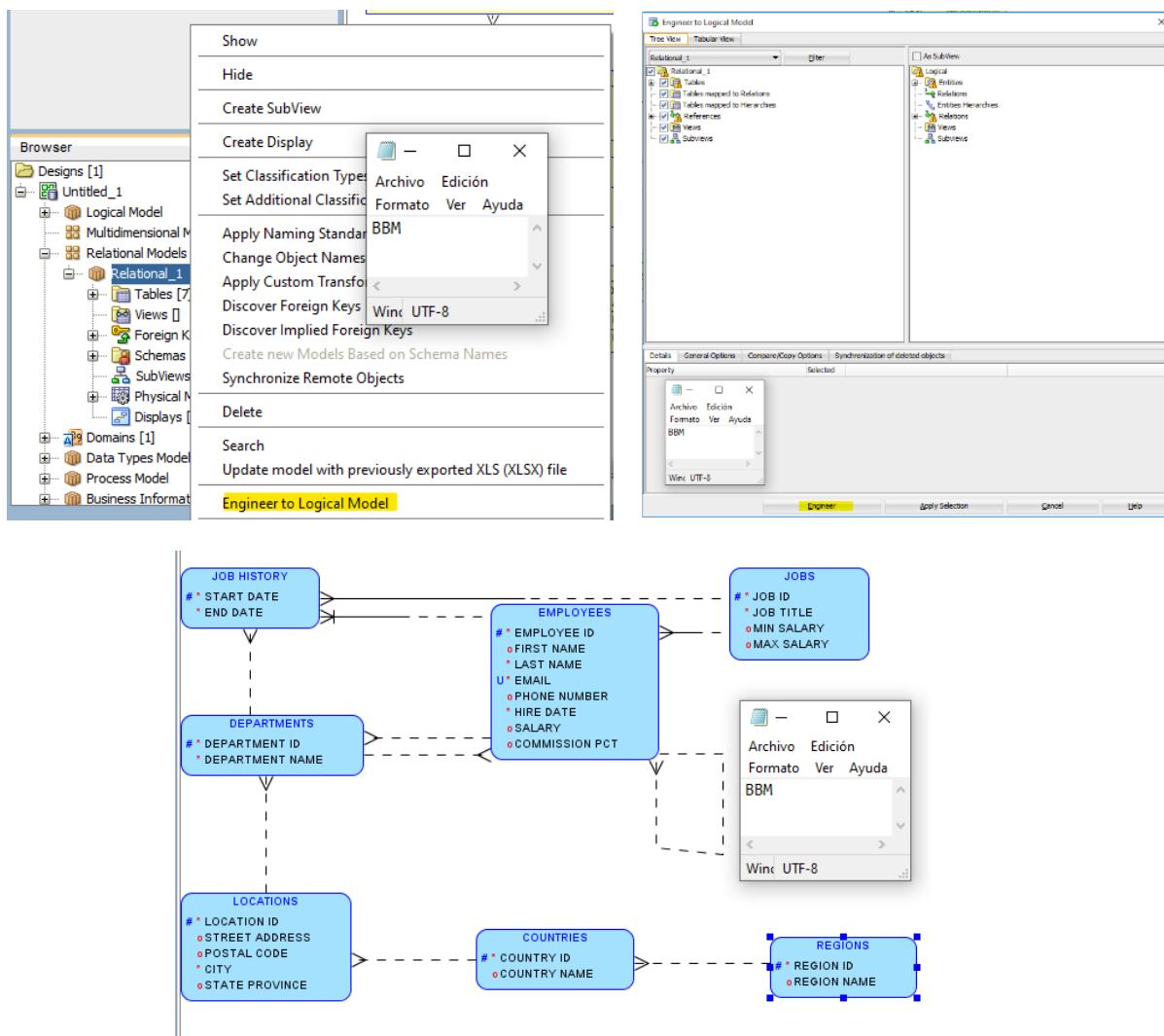
En la siguiente ventana me salen las tablas, selecciono todas y le doy a *Next*.



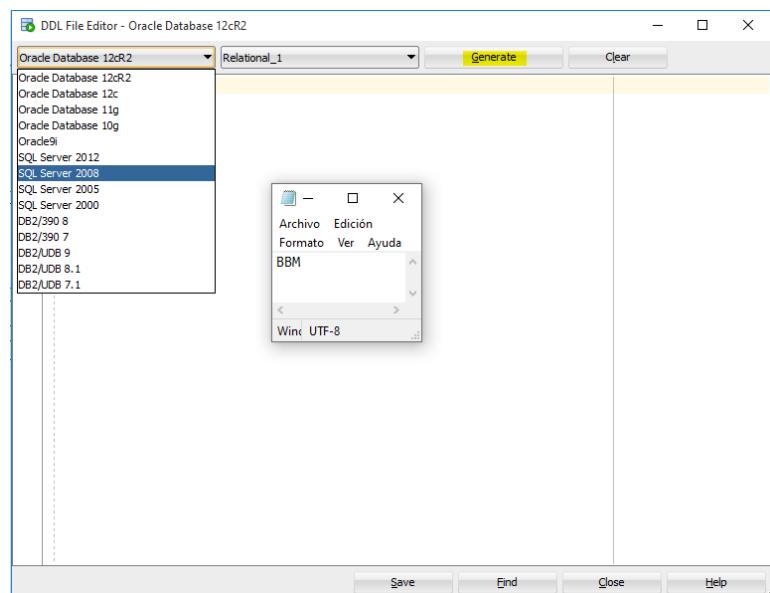
La última ventana es un resumen. Le doy a **Finish** y ya me presentaría el modelo relacional.



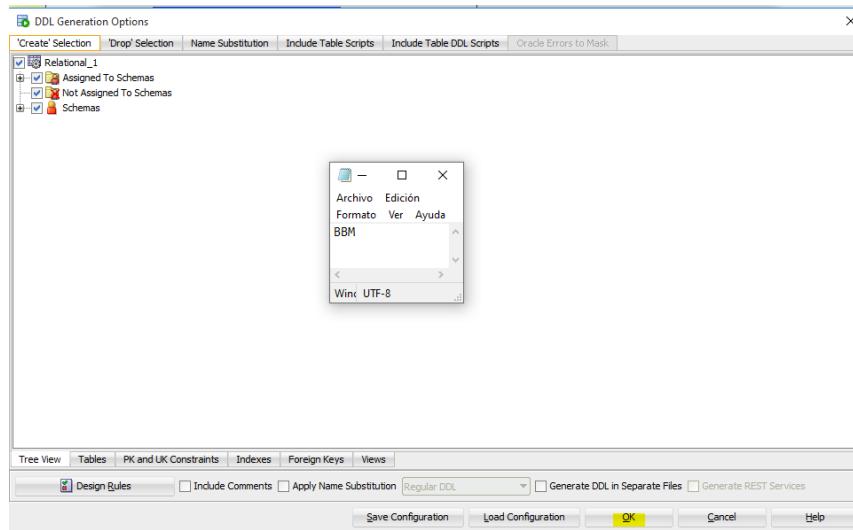
Ahora para hacer la ingeniería hacia el modelo lógico, hago clic en el botón y ya me sale el modelo (también lo puedo hacer en el *Browser* haciendo clic con el botón derecho sobre **Relational_1** y haciendo clic en **Engineer to Logical Model**).



Si estoy en el modelo relacional, puedo generar un **DDL** (Data Definition Language). ¿Cómo lo hago? Puedo pinchar en uno o coger todo para generar un *script*. En caso de no ver el *Browser*, me voy a **View>Data Modeler>Browser**. En él, hago click con en el ícono (Generate DDL). Me sale una ventana en la que tengo para elegir entre *Oracle*, *SQL* y *DB2*. Le digo que lo quiero en **SQL Server 2008**. Hago clic en el botón *Generate*.



Me sale una ventana con toda la información, la cual acepto dando a **OK** y me sale un *script* el cual puedo abrir en SQL Server.



DDL File Editor - SQL Server 2008

```

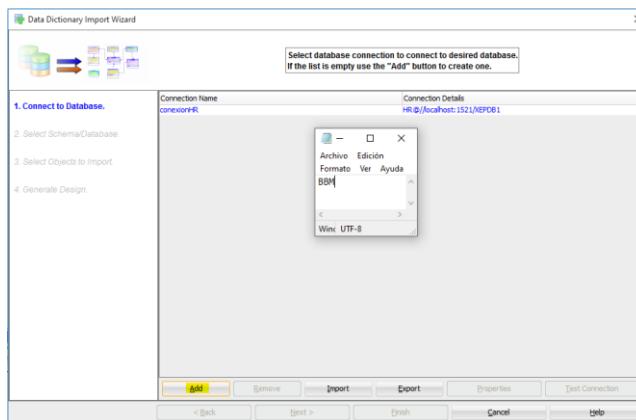
SQL Server 2008 Relational_1 Generate Clear
1 -- Generated by Oracle SQL Developer Data Modeler 20.2.0.167.1538
2 -- at: 2020-11-18 17:03:08 CET
3 -- site: SQL Server 2008
4 -- type: SQL Server 2008
5
6
7
8 CREATE TABLE HR.COUNTRIES
9 (
10     COUNTRY_ID CHAR (2) NOT NULL ,
11     COUNTRY_NAME VARCHAR (40) ,
12     REGION_ID NUMERIC (28)
13 )
14 GO
15
16
17
18 EXEC sp_addextendedproperty 'MS_Description' , 'country table. Contains 25 rows. References with local foreign keys.' , 'TABLE' , 'COUNTRIES'
19 GO
20
21
22
23 EXEC sp_addextendedproperty 'MS_Description' , 'Primary key of countries table.' , 'USER' , 'dbo' , 'COUNTRIES'
24 GO
25
26
27
28 EXEC sp_addextendedproperty 'MS_Description' , 'Country name' , 'USER' , 'dbo' , 'TABLE' , 'COUNTRIES'

```

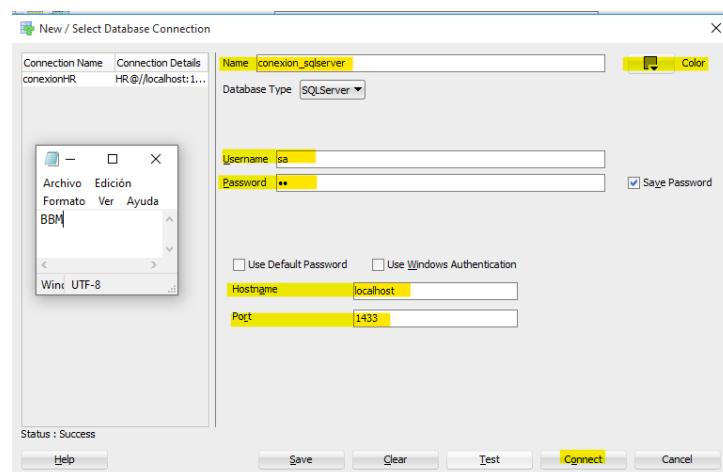
Ahora tratamos de hacer una Ingeniería desde el *Data Modeler* sobre la base de datos *Pubs*. Para acceder al *Data Modeler* desde *SQL Developer*, hago clic en la siguiente ruta **File>Data Modeler>Import>Data dictionary**.

NOTA: Como recordatorio, para poder hacer la conexión con *SQL Server*, hemos de ir a **Tools>Preferences>Data Modeler> Third Party JDBC Drivers**. Voy a **File>Import>Data Dictionary**, En la conexión tendría *SQL Server* en el apartado *Datatype*.

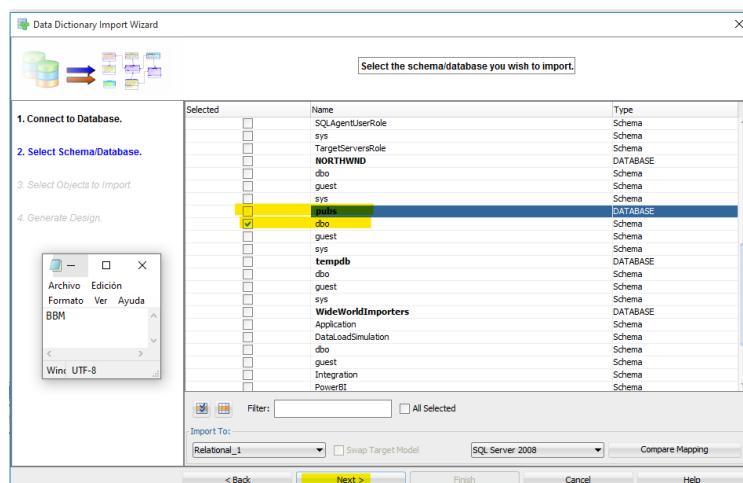
Hago en la conexión. La única diferencia con el *SQL Developer* es que en la primera pantalla hemos de darle al botón **Add** en vez de clicar sobre una de las bases que tenemos:



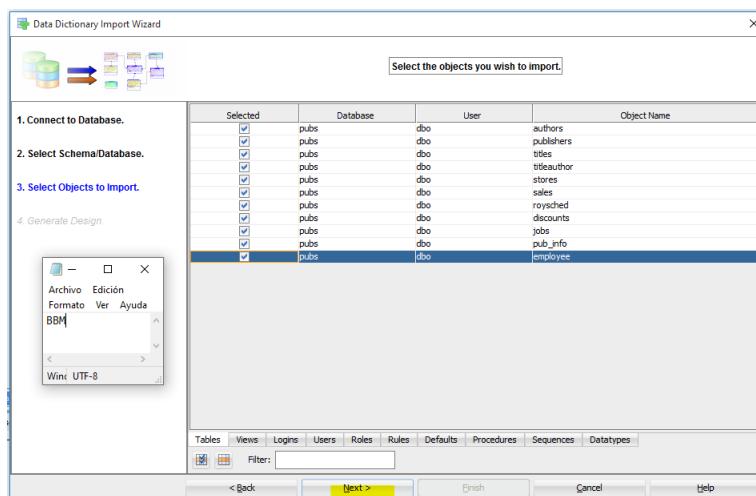
En las opciones cubro de la siguiente manera (especial atención en el usuario, que es **sa** con su contraseña **sa**). Cuando la tengo cubierta, le doy a *Next*.



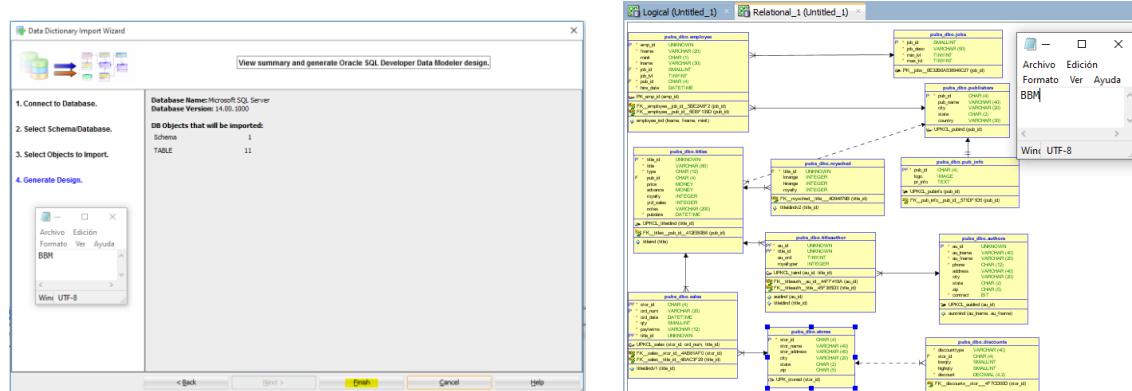
La siguiente ventana son los esquemas, de los cuales selecciono **Pubs** (en este caso me interesa elegir **dbo**. Quito **sys** y **guest**). Le volvemos a dar *Next* y aparecen las tablas.



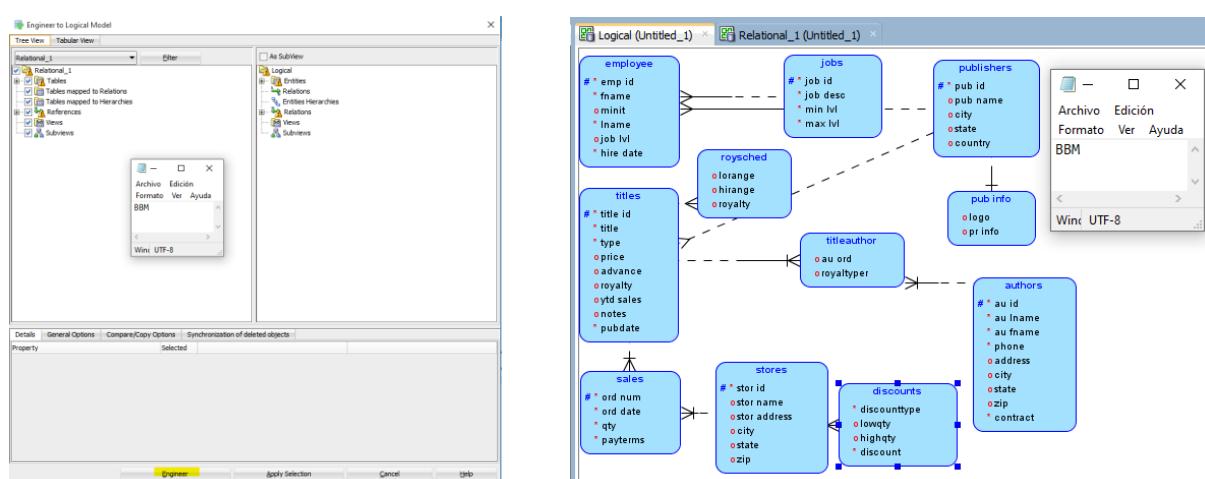
En la siguiente ventana me salen las tablas. Seleccióno todas y le doy a *Next*.



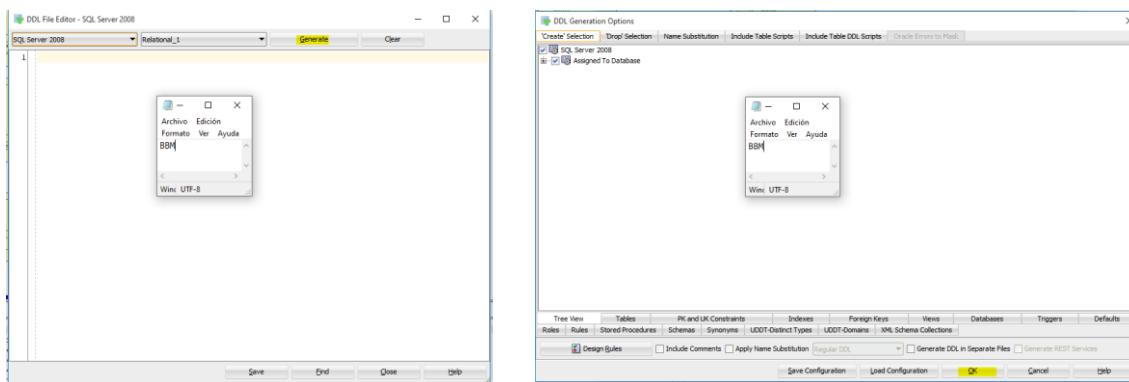
La última ventana es un resumen. Le doy a **Finish** y ya me presentaría el modelo relacional en *Data Modeler*.



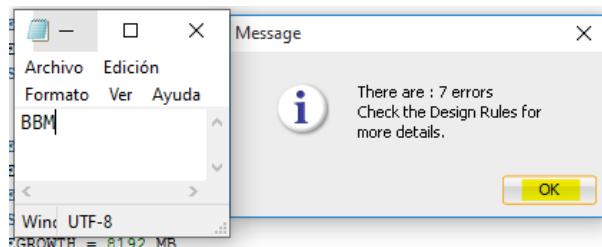
Ahora para hacer la ingeniería hacia el modelo lógico, hago clic en el botón (igual que antes). Ciclo en el botón **Engineer** y ya me sale el modelo.



Desde el *modelo relacional*, hago clic en el botón . Hago selección en *SQL Server 2008*.



NOTA: Hay un par de datos que salen en el *script* que hay que revisar porque suelen dar fallos.



```

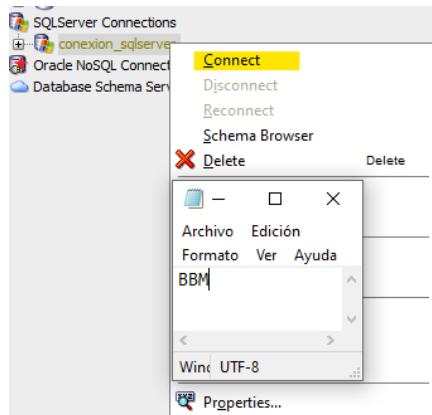
1 -- Generated by Oracle SQL Developer Data Modeler 20.3.0.283.0710
2 -- at: 2020-11-18 17:48:08 CET
3 -- site: SQL Server 2008
4 -- type: SQL Server 2008
5
6
7
8 CREATE DATABASE pubs
9   ON PRIMARY
10  (
11    NAME = 'DB1_Data.MDF',
12    FILENAME = 'c:\program files\microsoft sql server\mssql\data\DB1_Data.MDF',
13    MAXSIZE = UNLIMITED
14  ),
15  (
16    NAME = 'pubs',
17    FILENAME = 'C:\Program Files\Microsoft SQL Server\MSSQL14.MSSQLSERVER\MSSQL\DATA\pubs.mdf',
18    SIZE = 1024 MB,
19    MAXSIZE = UNLIMITED,
20    FILEGROWTH = 8192 MB
21  )
22 GO
23
24 ALTER DATABASE pubs
25   SET MULTI_USER ,
26   CURSOR_CLOSE_ON_COMMIT ON ,
27   CURSOR_DEFAULT GLOBAL ,
28   AUTO_CLOSE ON ,

```

Para acabar, vamos a insertar desde script una base de datos en SQL Datamodeler.

Ahora cogemos el *script* de nombre *sample_data* y voy a insertar desde script. Lo que hacemos es lanzar un *script* desde el **SQL Developer**. Copio y pego en el *schema* que tengo activo.

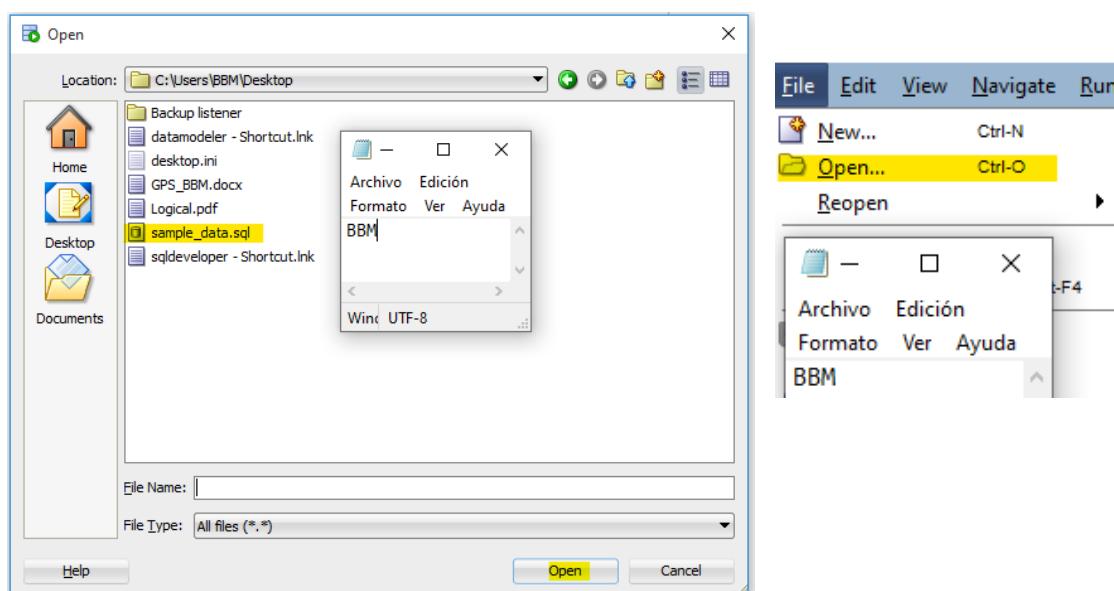
Primero hacemos doble clic sobre la *conexión_sqlserver* o bien botón derecho>Connect.



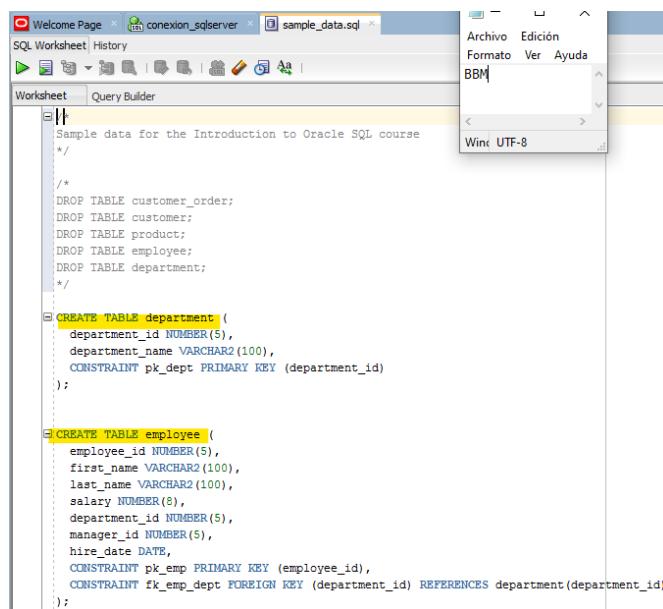
Se nos abre una pantalla para escribir un *script* a mayores de las bases de datos.



Le damos a **File>Open** y elegimos el archivo deseado. Le damos a **Open**.



Con esto ya se abriría el script para generar la base de datos. Pulsaríamos el botón Run script ➤ y tendríamos creada nuestra base de datos.



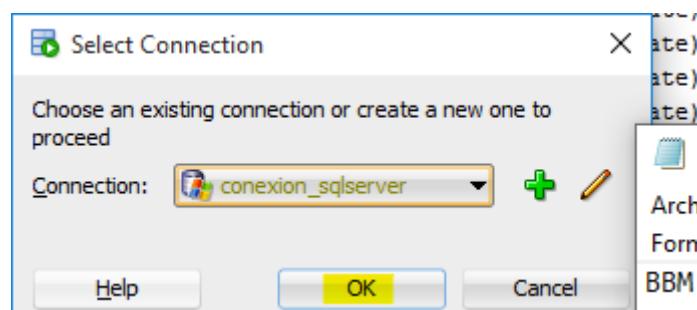
```
/* Sample data for the Introduction to Oracle SQL course */

/*
DROP TABLE customer_order;
DROP TABLE customer;
DROP TABLE product;
DROP TABLE employee;
DROP TABLE department;
*/

CREATE TABLE department (
    department_id NUMBER(5),
    department_name VARCHAR2(100),
    CONSTRAINT pk_dept PRIMARY KEY (department_id)
);

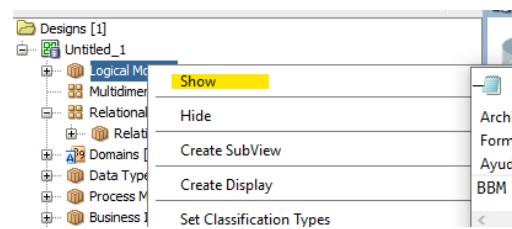
CREATE TABLE employee (
    employee_id NUMBER(5),
    first_name VARCHAR2(100),
    last_name VARCHAR2(100),
    salary NUMBER(8),
    department_id NUMBER(5),
    manager_id NUMBER(5),
    hire_date DATE,
    CONSTRAINT pk_emp PRIMARY KEY (employee_id),
    CONSTRAINT fk_emp_dept FOREIGN KEY (department_id) REFERENCES department(department_id)
);
```

NOTA: Cuando pulsamos el botón, nos pregunta en qué conexión queremos meterlo. Elegimos la *conexión de sql server*.

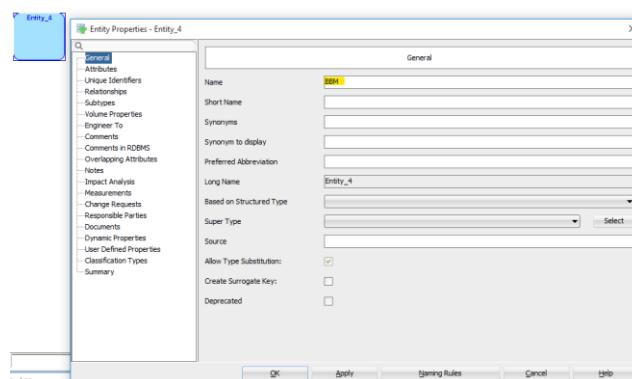


Modelado Lógico - Relacional - Físico Proyecto

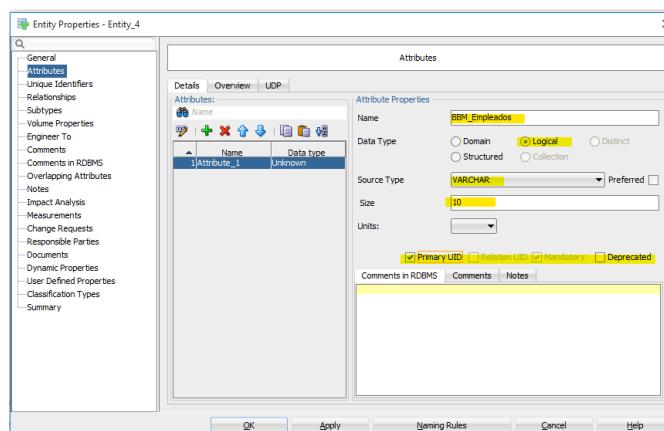
Explicado todo lo anterior, dejo registrado mi propio modelo la base de datos del proyecto. Para ello. En el data modeler, hacemos clic con el botón derecho en **Logical Model**, y le damos a **Show**:



Al igual que siempre, para crear el modelo, hacemos clic en el botón y manteniendo pulsado, arrastrando sobre la pantalla y soltando creamos las tablas que van a componer nuestro *modelo lógico*. Como consecuencia de ello, en la imagen que viene a continuación, vemos como queda creada una **entidad** con su correspondiente pop-up donde podremos **editar la tabla a nuestro antojo**. De todas estas opciones, yo editaré dos pestañas, la pestaña **General**, donde le pondré el **nombre** a la entidad que voy a crear:



También la pestaña **Attributes**, donde pondremos lo que serán las futuras **columnas de las tablas** con sus respectivos **tipos**. Para ellos, hacemos clic en el botón y se despliegan unas opciones donde elegiremos el nombre del *atributo* y el tipo. A mayores le daremos identificadores como puede ser **Primary Key**:

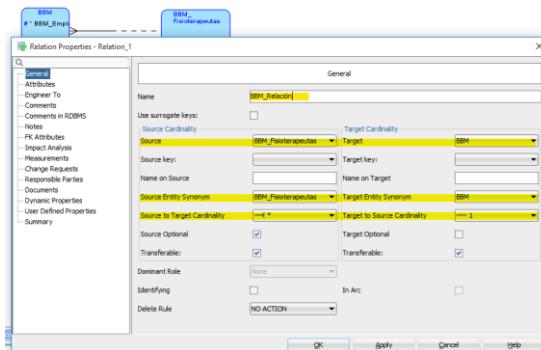


Para hacer una **relación** entre las dos, elegiré clicando en uno de estos cuatro botones si quiero relaciones **1|1**, **1|n**, **n|m**. El proceso es igual que para

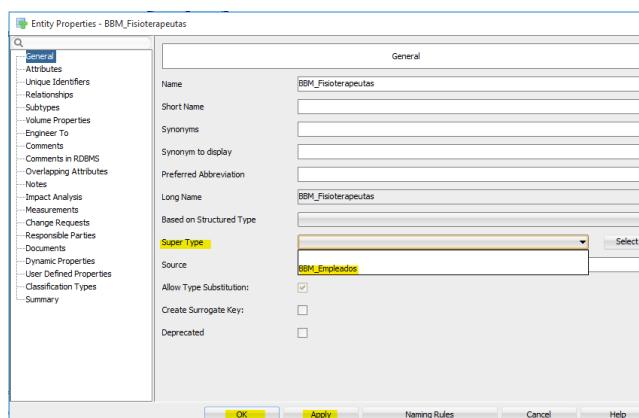
crear la entidad: hacemos clic con el botón izquierdo en uno de los botones mencionados anteriormente, y pinchando primero en una y luego en la otra:



Así, se abre una ventana emergente donde podremos entre otras cosas aportarle un nombre a esa relación, o editar en caso de equivocarnos cuál es cuál:



Volviendo a la pestaña *General*, en función de si las entidades que creo son **subtipos** o no, también podemos crearlos. Haciendo clic en la entidad que va a ser subtipo y desplegando la ventana, hay una opción que se llama **Super Type**, en donde elegimos la entidad que va a ser el **supertipo**.

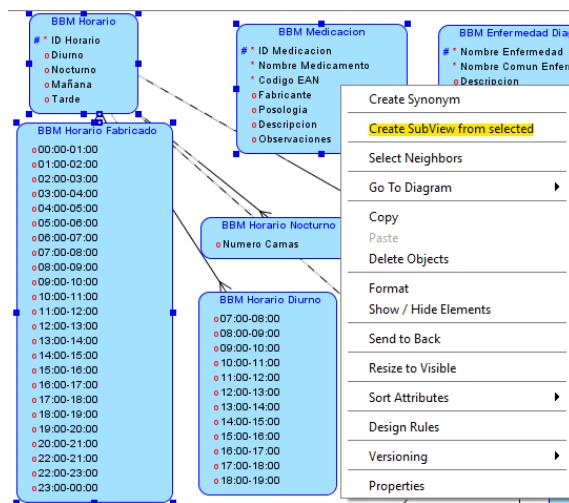


De esta forma, una entidad queda encuadrada en la otra. Tan solo hemos de hacer clic con el botón derecho en **Realice to Visible** para poder ver ambas:

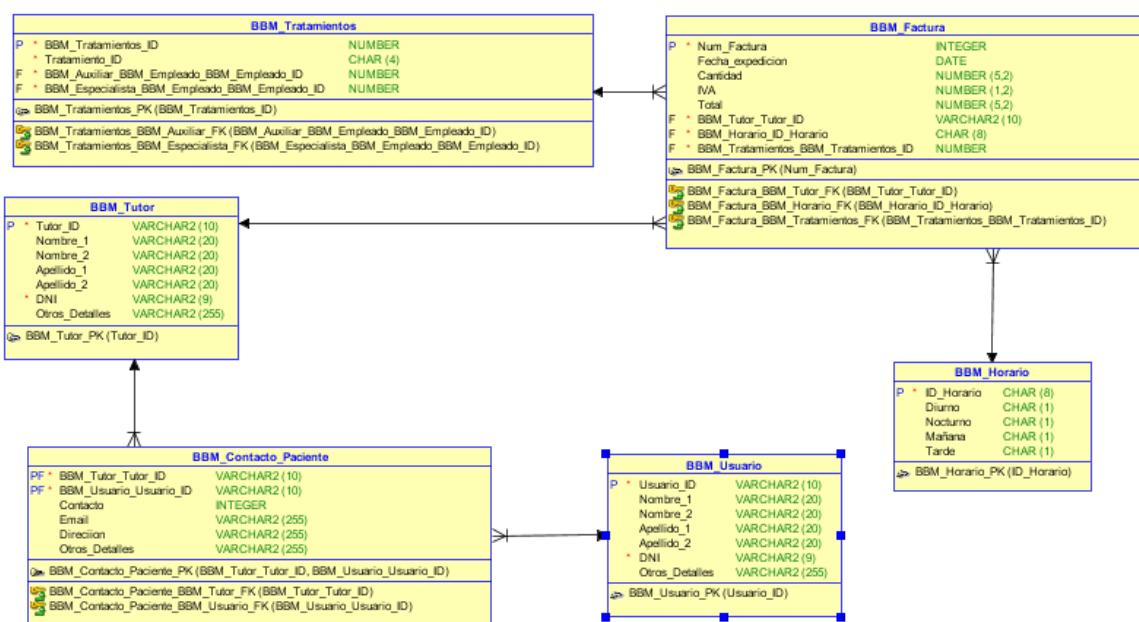


Subvistas

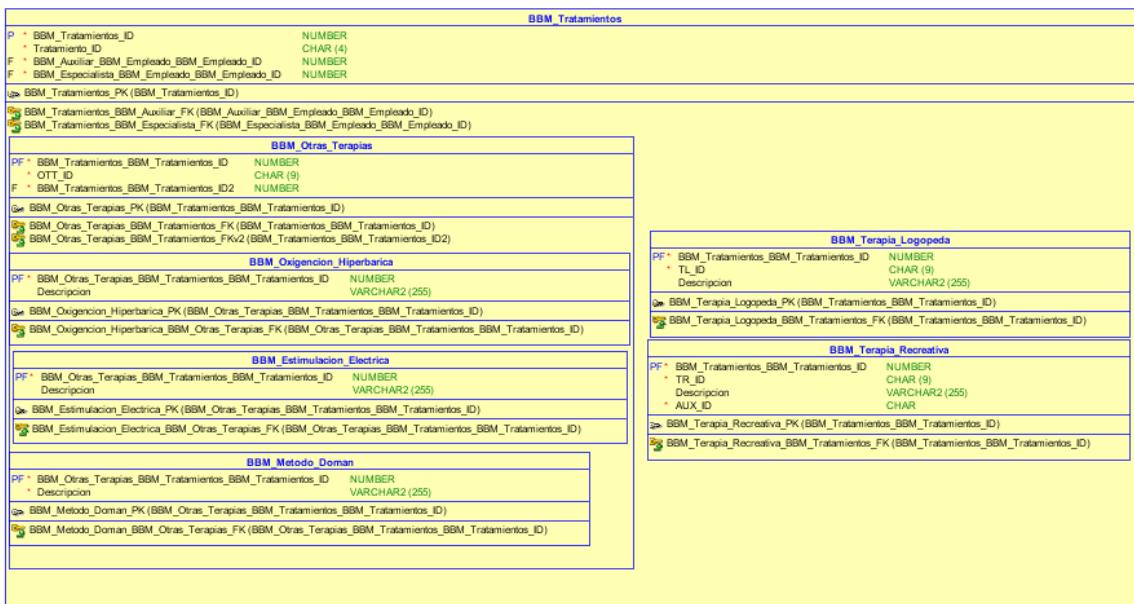
Con esta breve explicación, pasamos al modelado en sí. Como ya expliqué en la [introducción](#), la **base de datos sobre lo que versa mi proyecto** es sobre los **centros de cuidados de personas con diversidad funcional**, y para focalizarlo más, centrado en **ASPACE**. Para la presentación en sí, he dividido en **subvistas** el proyecto para hacer una explicación más sencilla y por partes. ¿Cómo se hace una subvista?, seleccionando varias tablas y con la tecla **CTRL + clic** de botón izquierdo sobre la entidad. Al hacer esto, pulsamos **clic de botón derecho** y le damos a **Create SubView from selected**:



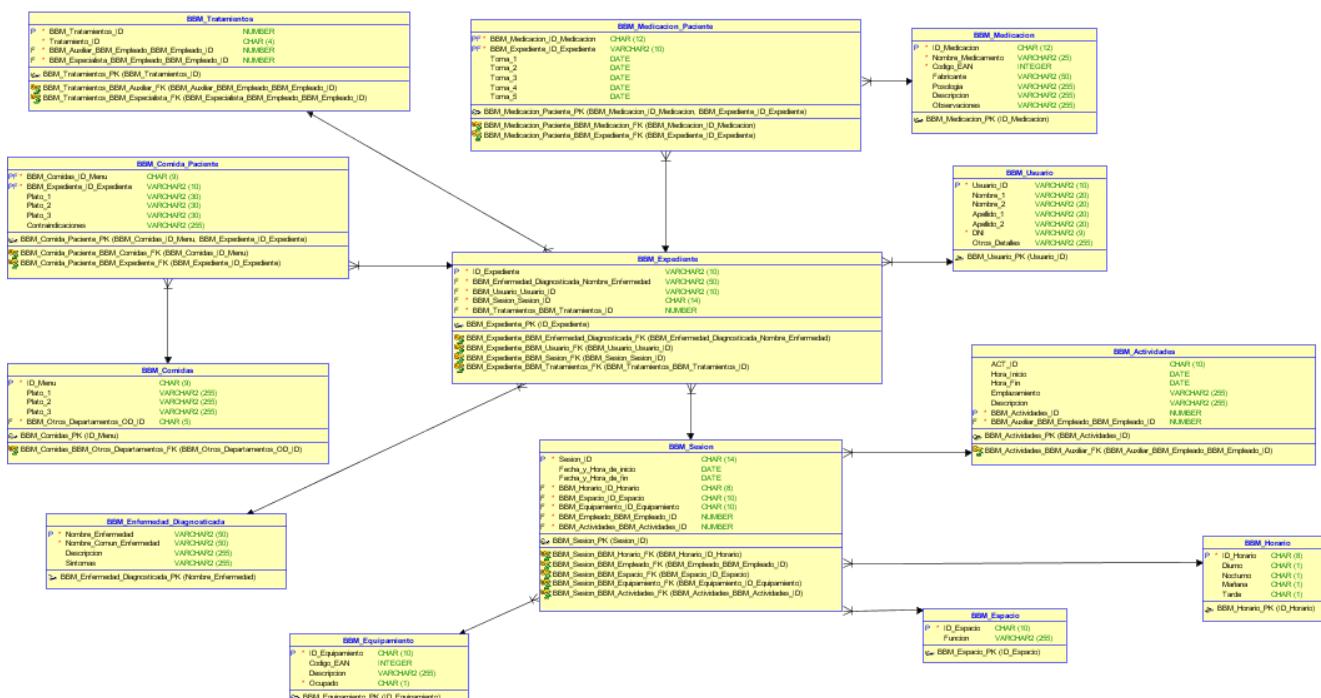
De todas estas vistas, yo he creado varias para que el conjunto luego tenga una comprensión legible. En esta línea, la primera *subvista* trata sobre la parte de los usuarios. Para que las cosas no se compliquen, la tabla **usuarios** es uno de los actores principales, uniéndose a su **tutor** mediante una **relación n|m**, que da como resultado una tercera tabla que será el **Contacto del paciente**. Quien paga las **facturas** de los usuarios son los tutores, y esas facturas están basadas en los **tratamientos** que reciben y el **horario** (si es nocturno o diurno):



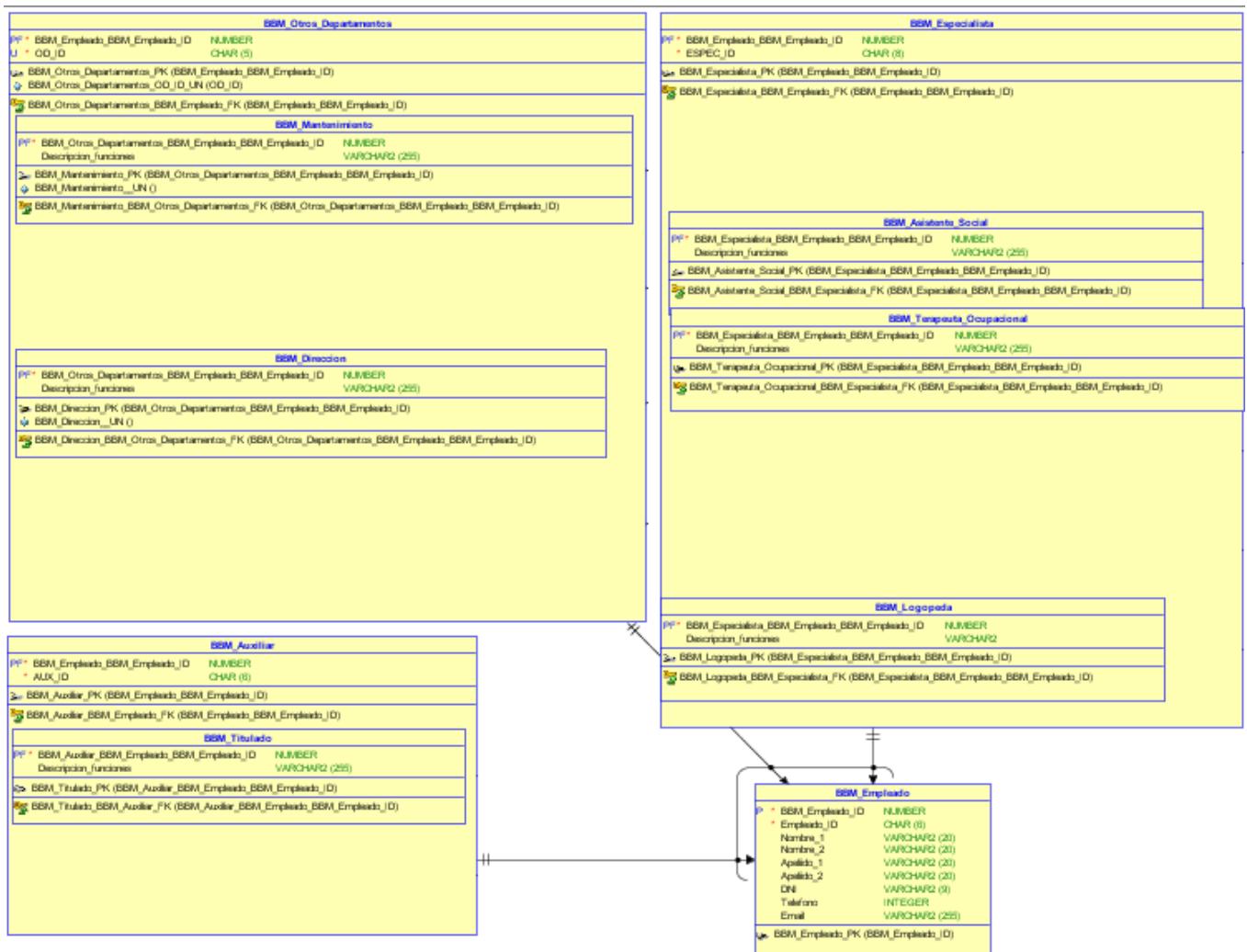
Ni que decir tiene que los **tratamientos** que reciben son diversos, aquí dejo registrada la tabla con todos y cada uno de los que hay:



Todos los tratamientos se les administra en **sesiones** (como pueden ser individuales o grupales, pero se cobra individual yo he hecho una sola tabla al respecto). Dichas sesiones quedan registradas en un **expediente** donde se incluyen: las **comidas** que puede tomar el paciente, la **enfermedad diagnosticada** y la **Medicación** que toma.

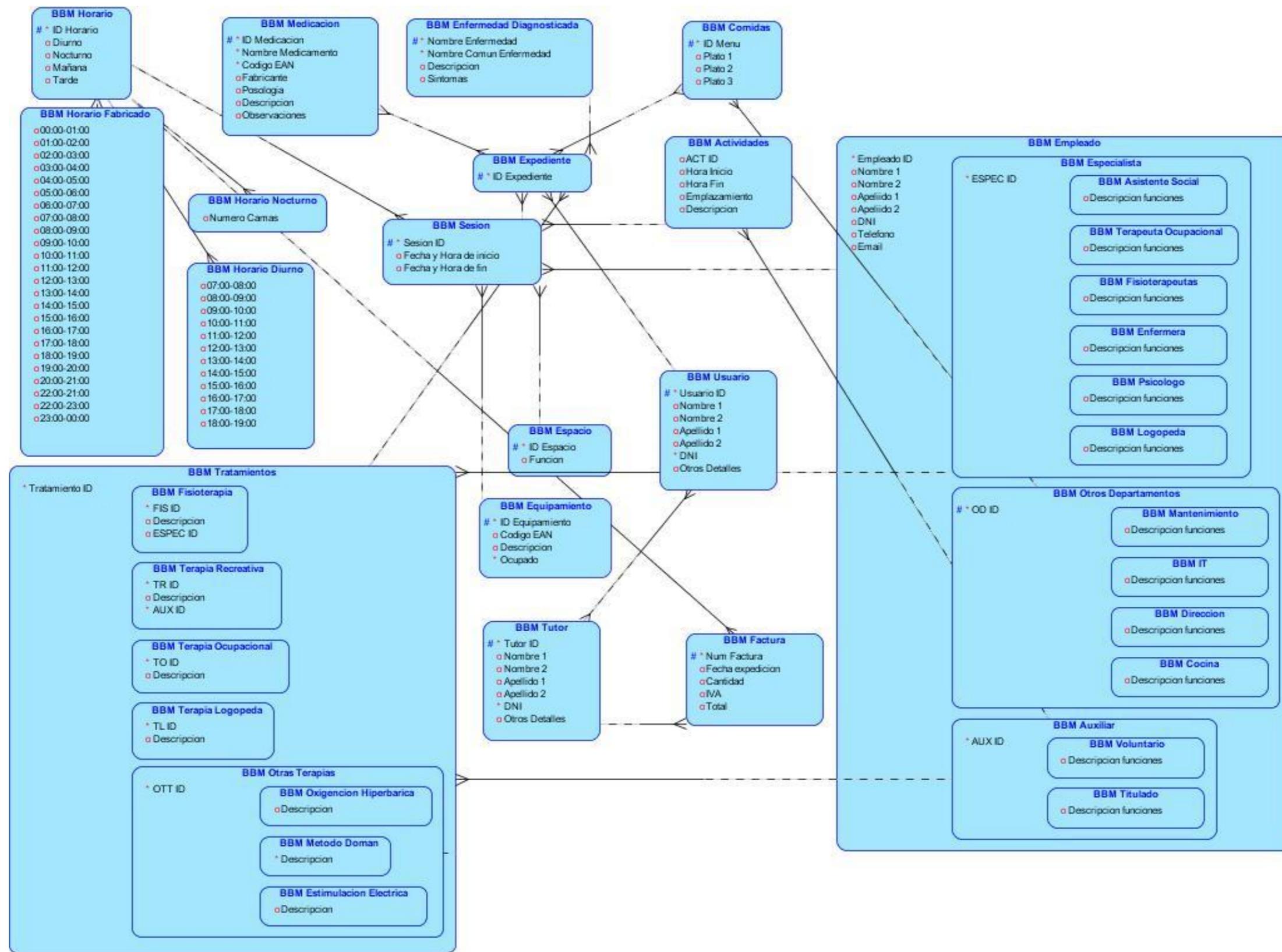


Todo esto que he explicado anteriormente, está organizado y tratado por los **empleados**, de los cuales cuentan con **especialistas**, con **auxiliares** (titulados o no) y con **otros departamentos** para hacer esto funcional:

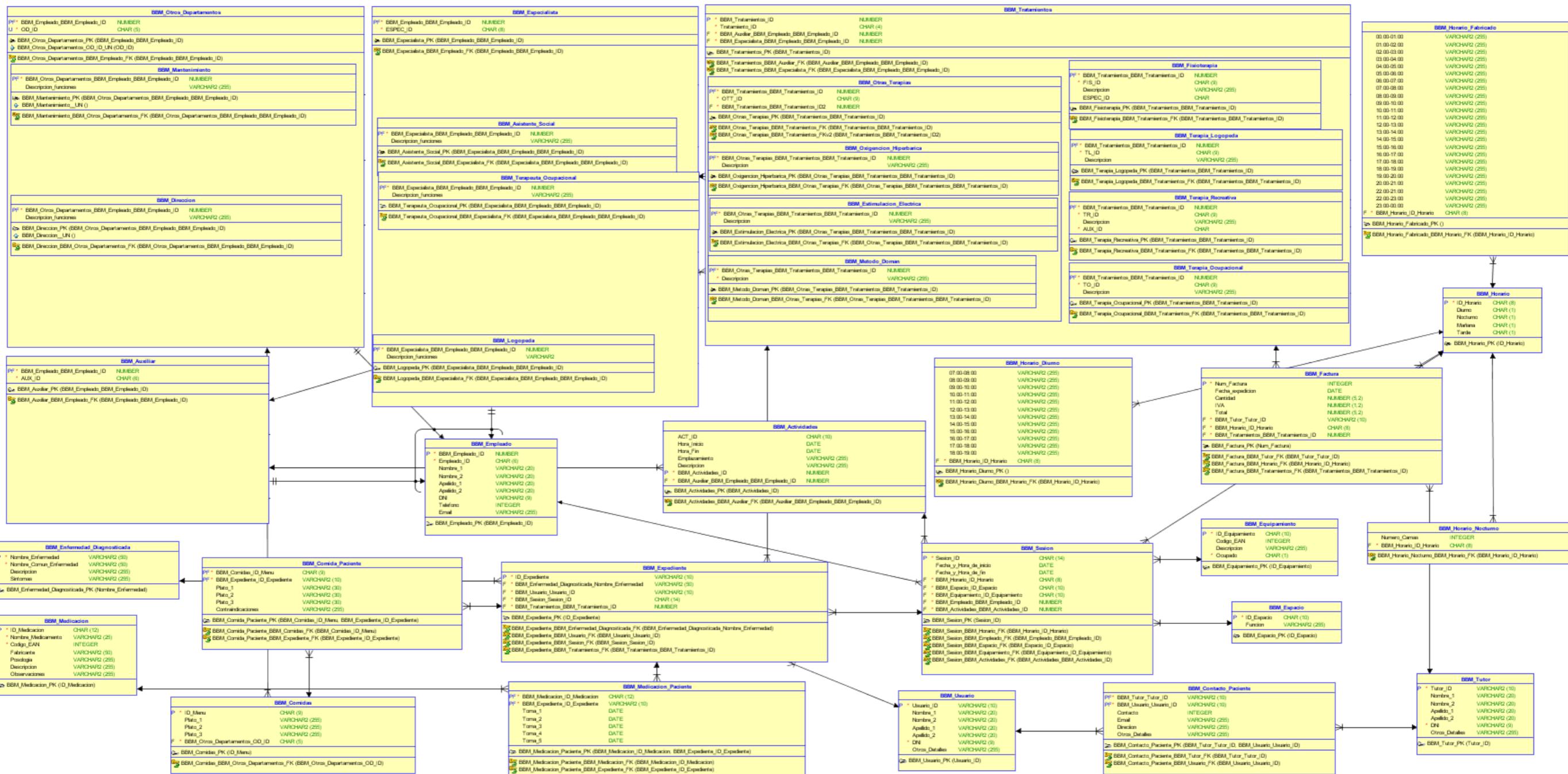


A continuación, dejo registro de los modelos lógico y relacional para posteriormente dejar su paso a exportarlo a **SQL Server**:

Modelo Lógico

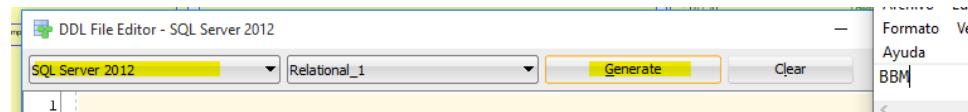


Modelo Relacional

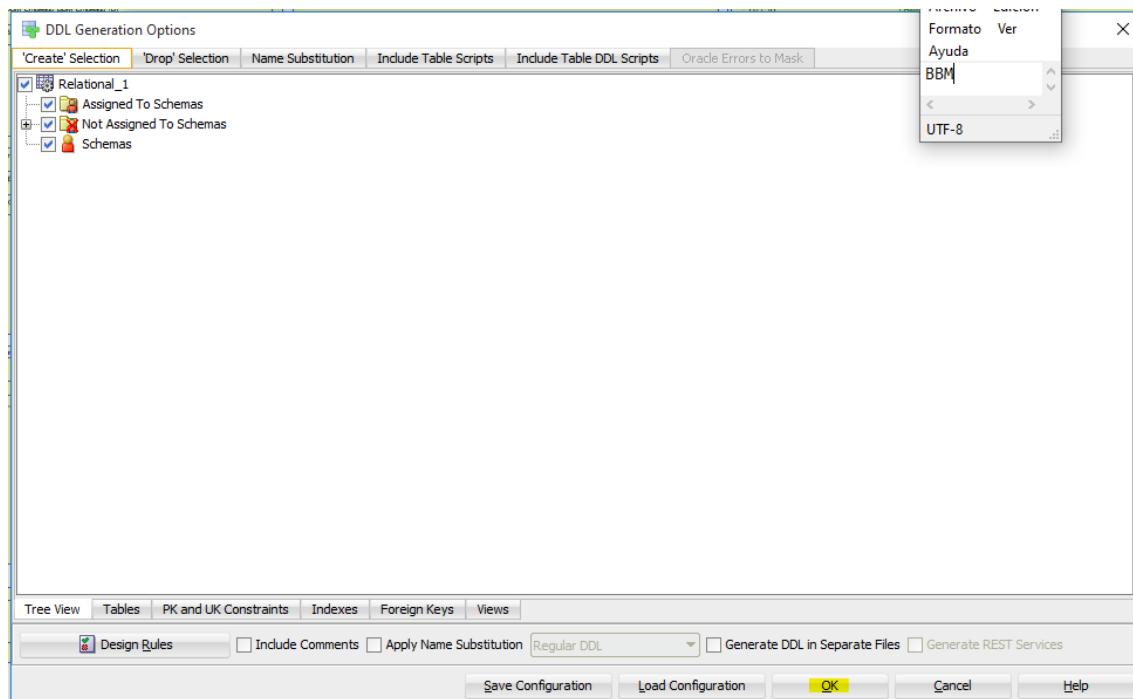


Generar Archivo DDL

Para poder exportar el archivo a SQL Server, hay que generar un archivo en formato **.ddl**. Para ello, hacemos clic en el botón  . Al hacerlo, se abre una nueva ventana donde elegiremos el formato en que lo queremos sacar (SQL Server 2012), y le daremos al botón **Generate**:



Al darle a Generate, se abre una ventana emergente, aceptamos:



Vemos que se genera el siguiente script:

```
-- Generated by Oracle SQL Developer Data Modeler 20.3.0.283.0710
-- at: 2021-01-23 19:49:44 CET
-- site: SQL Server 2012
-- type: SQL Server 2012
```

```
USE BBM_ASPACE
GO
```

```
CREATE TABLE BBM_Actividades
(
    ACT_ID CHAR (10) ,
    Hora_Inicio DATETIME ,
    Hora_Fin DATETIME ,
```

```
        Emplazamiento VARCHAR (255) ,
        Descripcion VARCHAR (255) ,
        BBM_Actividades_ID INTEGER NOT NULL IDENTITY NOT FOR REPLICATION ,
        BBM_Auxiliar_BBM_Empleado_BBM_Empleado_ID INTEGER NOT NULL
    )
GO

ALTER TABLE BBM_Actividades ADD CONSTRAINT BBM_Actividades_PK PRIMARY KEY
CLUSTERED (BBM_Actividades_ID)
    WITH (
        ALLOW_PAGE_LOCKS = ON ,
        ALLOW_ROW_LOCKS = ON )
GO

CREATE TABLE BBM_Asistente_Social
(
    BBM_Especialista_BBM_Empleado_BBM_Empleado_ID INTEGER NOT NULL ,
    Descripcion_funciones VARCHAR (255)
)
GO

ALTER TABLE BBM_Asistente_Social ADD CONSTRAINT BBM_Asistente_Social_PK
PRIMARY KEY CLUSTERED (BBM_Especialista_BBM_Empleado_BBM_Empleado_ID)
    WITH (
        ALLOW_PAGE_LOCKS = ON ,
        ALLOW_ROW_LOCKS = ON )
GO

CREATE TABLE BBM_Auxiliar
(
    BBM_Empleado_BBM_Empleado_ID INTEGER NOT NULL ,
    AUX_ID CHAR (6) NOT NULL
)
GO

ALTER TABLE BBM_Auxiliar ADD CONSTRAINT BBM_Auxiliar_PK PRIMARY KEY
CLUSTERED (BBM_Empleado_BBM_Empleado_ID)
    WITH (
        ALLOW_PAGE_LOCKS = ON ,
        ALLOW_ROW_LOCKS = ON )
GO

CREATE TABLE BBM_Cocina
(
    BBM_Otros_Departamentos_BBM_Empleado_BBM_Empleado_ID INTEGER NOT
NULL ,
    Descripcion_funciones VARCHAR (255)
)
GO

ALTER TABLE BBM_Cocina ADD CONSTRAINT BBM_Cocina_PK PRIMARY KEY CLUSTERED
(BBM_Otros_Departamentos_BBM_Empleado_BBM_Empleado_ID)
    WITH (
        ALLOW_PAGE_LOCKS = ON ,
        ALLOW_ROW_LOCKS = ON )
GO
```

```
CREATE TABLE BBM_Comida_Paciente
(
    BBM_Comidas_ID_Menu CHAR (9) NOT NULL ,
    BBM_Expediente_ID_Expediente VARCHAR (10) NOT NULL ,
    Plato_1 VARCHAR (30) ,
    Plato_2 VARCHAR (30) ,
    Plato_3 VARCHAR (30) ,
    Contraindicaciones VARCHAR (255)
)
GO

ALTER TABLE BBM_Comida_Paciente ADD CONSTRAINT BBM_Comida_Paciente_PK
PRIMARY KEY CLUSTERED (BBM_Comidas_ID_Menu, BBM_Expediente_ID_Expediente)
    WITH (
        ALLOW_PAGE_LOCKS = ON ,
        ALLOW_ROW_LOCKS = ON )
GO

CREATE TABLE BBM_Comidas
(
    ID_Menu CHAR (9) NOT NULL ,
    Plato_1 VARCHAR (255) ,
    Plato_2 VARCHAR (255) ,
    Plato_3 VARCHAR (255) ,
    BBM_Otros_Departamentos_OD_ID CHAR (5) NOT NULL
)
GO

EXEC sp_addextendedproperty 'MS_Description' , 'Dde la semana' , 'USER' ,
'dbo' , 'TABLE' , 'BBM_Comidas' , 'COLUMN' , 'ID_Menu'
GO

ALTER TABLE BBM_Comidas ADD CONSTRAINT BBM_Comidas_PK PRIMARY KEY
CLUSTERED (ID_Menu)
    WITH (
        ALLOW_PAGE_LOCKS = ON ,
        ALLOW_ROW_LOCKS = ON )
GO

CREATE TABLE BBM_Contacto_Paciente
(
    BBM_Tutor_Tutor_ID VARCHAR (10) NOT NULL ,
    BBM_Usuario_Usuario_ID VARCHAR (10) NOT NULL ,
    Contacto INTEGER ,
    Email VARCHAR (255) ,
    Direccion VARCHAR (255) ,
    Otros_Detalles VARCHAR (255)
)
GO

ALTER TABLE BBM_Contacto_Paciente ADD CONSTRAINT BBM_Contacto_Paciente_PK
PRIMARY KEY CLUSTERED (BBM_Tutor_Tutor_ID, BBM_Usuario_Usuario_ID)
    WITH (
        ALLOW_PAGE_LOCKS = ON ,
        ALLOW_ROW_LOCKS = ON )
```

```
GO

CREATE TABLE BBM_Direccion
(
    BBM_Otros_Departamentos_BBM_Emppleado_BBM_Emppleado_ID INTEGER NOT
NULL ,
    Descripcion_funciones VARCHAR (255)
)
GO

ALTER TABLE BBM_Direccion ADD CONSTRAINT BBM_Direccion_PK PRIMARY KEY
CLUSTERED (BBM_Otros_Departamentos_BBM_Emppleado_BBM_Emppleado_ID)
    WITH (
        ALLOW_PAGE_LOCKS = ON ,
        ALLOW_ROW_LOCKS = ON )
GO

CREATE TABLE BBM_Empleado
(
    BBM_Empleado_ID INTEGER NOT NULL IDENTITY NOT FOR REPLICATION ,
    Empleado_ID CHAR (6) NOT NULL ,
    Nombre_1 VARCHAR (20) ,
    Nombre_2 VARCHAR (20) ,
    Apellido_1 VARCHAR (20) ,
    Apellido_2 VARCHAR (20) ,
    DNI VARCHAR (9) ,
    Telefono INTEGER ,
    Email VARCHAR (255)
)
GO

EXEC sp_addextendedproperty 'MS_Description' , 'Iniciales + 3 ultimos
digitos DNI' , 'USER' , 'dbo' , 'TABLE' , 'BBM_Empleado' , 'COLUMN' ,
'Empleado_ID'
GO

ALTER TABLE BBM_Empleado ADD CONSTRAINT BBM_Empleado_PK PRIMARY KEY
CLUSTERED (BBM_Empleado_ID)
    WITH (
        ALLOW_PAGE_LOCKS = ON ,
        ALLOW_ROW_LOCKS = ON )
GO

CREATE TABLE BBM_Enfermedad_Diagnosticada
(
    Nombre_Enfermedad VARCHAR (50) NOT NULL ,
    Nombre_Comun_Enfermedad VARCHAR (50) NOT NULL ,
    Descripcion VARCHAR (255) ,
    Sintomas VARCHAR (255)
)
GO

ALTER TABLE BBM_Enfermedad_Diagnosticada ADD CONSTRAINT
BBM_Enfermedad_Diagnosticada_PK PRIMARY KEY CLUSTERED (Nombre_Enfermedad)
    WITH (
```

```

        ALLOW_PAGE_LOCKS = ON ,
        ALLOW_ROW_LOCKS = ON )
GO

CREATE TABLE BBM_Enfermera
(
    BBM_Especialista_BBM_Empleado_BBM_Empleado_ID INTEGER NOT NULL ,
    Descripcion_funciones VARCHAR
)
GO

ALTER TABLE BBM_Enfermera ADD CONSTRAINT BBM_Enfermera_PK PRIMARY KEY
CLUSTERED (BBM_Especialista_BBM_Empleado_BBM_Empleado_ID)
    WITH (
        ALLOW_PAGE_LOCKS = ON ,
        ALLOW_ROW_LOCKS = ON )
GO

CREATE TABLE BBM_Equipamiento
(
    ID_Equipamiento CHAR (10) NOT NULL ,
    Codigo_EAN INTEGER ,
    Descripcion VARCHAR (255) ,
    Ocupado BIT NOT NULL
)
GO

ALTER TABLE BBM_Equipamiento ADD CONSTRAINT BBM_Equipamiento_PK PRIMARY
KEY CLUSTERED (ID_Equipamiento)
    WITH (
        ALLOW_PAGE_LOCKS = ON ,
        ALLOW_ROW_LOCKS = ON )
GO

CREATE TABLE BBM_Espacio
(
    ID_Espacio CHAR (10) NOT NULL ,
    Funcion VARCHAR (255)
)
GO

```

**EXEC sp_addextendedproperty 'MS_Description' , 'En funci e si es un
aula, una habitaci n una sala o despacho ser **

AUL
HAB
+ FUNCIM + NUMERO
SAL
DES

EJEMPLO:

```

AULTRAT010' , 'USER' , 'dbo' , 'TABLE' , 'BBM_Espacio' , 'COLUMN' ,
'ID_Espacio'
GO

```

```
ALTER TABLE BBM_Espacio ADD CONSTRAINT BBM_Espacio_PK PRIMARY KEY
CLUSTERED (ID_Espacio)
    WITH (
        ALLOW_PAGE_LOCKS = ON ,
        ALLOW_ROW_LOCKS = ON )
GO

CREATE TABLE BBM_Especialista
(
    BBM_Empleado_BBM_Empleado_ID INTEGER NOT NULL ,
    ESPEC_ID CHAR (8) NOT NULL
)
GO

ALTER TABLE BBM_Especialista ADD CONSTRAINT BBM_Especialista_PK PRIMARY
KEY CLUSTERED (BBM_Empleado_BBM_Empleado_ID)
    WITH (
        ALLOW_PAGE_LOCKS = ON ,
        ALLOW_ROW_LOCKS = ON )
GO

CREATE TABLE BBM_Estimulacion_Electrica
(
    BBM_Otras_Terapias_BBM_Tratamientos_BBM_Tratamientos_ID INTEGER NOT
NULL ,
    Descripcion VARCHAR (255)
)
GO

ALTER TABLE BBM_Estimulacion_Electrica ADD CONSTRAINT
BBM_Estimulacion_Electrica_PK PRIMARY KEY CLUSTERED
(BBM_Otras_Terapias_BBM_Tratamientos_BBM_Tratamientos_ID)
    WITH (
        ALLOW_PAGE_LOCKS = ON ,
        ALLOW_ROW_LOCKS = ON )
GO

CREATE TABLE BBM_Expediente
(
    ID_Expediente VARCHAR (10) NOT NULL ,
    BBM_Enfermedad_Diagnosticada_Nombre_Enfermedad VARCHAR (50) NOT NULL
,
    BBM_Usuario_Usuario_ID VARCHAR (10) NOT NULL ,
    BBM_Sesion_Sesion_ID CHAR (14) NOT NULL ,
    BBM_Tratamientos_BBM_Tratamientos_ID INTEGER NOT NULL
)
GO

EXEC sp_addextendedproperty 'MS_Description' , 'Compuesto por EXP +
Apellido + Apellido + Nombre Usuario + Numeracion
Ejemplo: EXPBBM0001' , 'USER' , 'dbo' , 'TABLE' , 'BBM_Expediente' ,
'COLUMN' , 'ID_Expediente'
GO
```

```
ALTER TABLE BBM_Expediente ADD CONSTRAINT BBM_Expediente_PK PRIMARY KEY
CLUSTERED (ID_Expediente)
    WITH (
        ALLOW_PAGE_LOCKS = ON ,
        ALLOW_ROW_LOCKS = ON )
GO

CREATE TABLE BBM_Factura
(
    Num_Factura INTEGER NOT NULL ,
    Fecha_expedicion DATE ,
    Cantidad NUMERIC (5,2) ,
    IVA NUMERIC (1,1) ,
    Total NUMERIC (5,2) ,
    BBM_Tutor_Tutor_ID VARCHAR (10) NOT NULL ,
    BBM_Horario_ID_Horario CHAR (8) NOT NULL ,
    BBM_Tratamientos_BBM_Tratamientos_ID INTEGER NOT NULL
)
GO

ALTER TABLE BBM_Factura ADD CONSTRAINT BBM_Factura_PK PRIMARY KEY
CLUSTERED (Num_Factura)
    WITH (
        ALLOW_PAGE_LOCKS = ON ,
        ALLOW_ROW_LOCKS = ON )
GO

CREATE TABLE BBM_Fisioterapeutas
(
    BBM_Especialista_BBM_Emppleado_BBM_Emppleado_ID INTEGER NOT NULL ,
    Descripcion_funciones VARCHAR (255)
)
GO

ALTER TABLE BBM_Fisioterapeutas ADD CONSTRAINT BBM_Fisioterapeutas_PK
PRIMARY KEY CLUSTERED (BBM_Especialista_BBM_Emppleado_BBM_Emppleado_ID)
    WITH (
        ALLOW_PAGE_LOCKS = ON ,
        ALLOW_ROW_LOCKS = ON )
GO

CREATE TABLE BBM_Fisioterapia
(
    BBM_Tratamientos_BBM_Tratamientos_ID INTEGER NOT NULL ,
    FIS_ID CHAR (9) NOT NULL ,
    Descripcion VARCHAR (255) ,
    ESPEC_ID CHAR
)
GO

ALTER TABLE BBM_Fisioterapia ADD CONSTRAINT BBM_Fisioterapia_PK PRIMARY
KEY CLUSTERED (BBM_Tratamientos_BBM_Tratamientos_ID)
    WITH (
        ALLOW_PAGE_LOCKS = ON ,
        ALLOW_ROW_LOCKS = ON )
GO
```

```
CREATE TABLE BBM_Horario
(
    ID_Horario CHAR (8) NOT NULL ,
    Diurno BIT ,
    Nocturno BIT ,
    Ma pan>ana BIT ,
    Tarde BIT
)
GO

EXEC sp_addextendedproperty 'MS_Description' , 'Para distinguir de
Nocturno, diurno o media jornada' , 'USER' , 'dbo' , 'TABLE' ,
'BBM_Horario' , 'COLUMN' , 'ID_Horario'
GO

ALTER TABLE BBM_Horario ADD CONSTRAINT BBM_Horario_PK PRIMARY KEY
CLUSTERED (ID_Horario)
    WITH (
        ALLOW_PAGE_LOCKS = ON ,
        ALLOW_ROW_LOCKS = ON )
GO

CREATE TABLE BBM_Horario_Diurno
(
    "07:00-08:00" VARCHAR (255) ,
    "08:00-09:00" VARCHAR (255) ,
    "09:00-10:00" VARCHAR (255) ,
    "10:00-11:00" VARCHAR (255) ,
    "11:00-12:00" VARCHAR (255) ,
    "12:00-13:00" VARCHAR (255) ,
    "13:00-14:00" VARCHAR (255) ,
    "14:00-15:00" VARCHAR (255) ,
    "15:00-16:00" VARCHAR (255) ,
    "16:00-17:00" VARCHAR (255) ,
    "17:00-18:00" VARCHAR (255) ,
    "18:00-19:00" VARCHAR (255) ,
    BBM_Horario_ID_Horario CHAR (8) NOT NULL
)
GO

CREATE TABLE BBM_Horario_Fabricado
(
    "00:00-01:00" VARCHAR (255) ,
    "01:00-02:00" VARCHAR (255) ,
    "02:00-03:00" VARCHAR (255) ,
    "03:00-04:00" VARCHAR (255) ,
    "04:00-05:00" VARCHAR (255) ,
    "05:00-06:00" VARCHAR (255) ,
    "06:00-07:00" VARCHAR (255) ,
    "07:00-08:00" VARCHAR (255) ,
    "08:00-09:00" VARCHAR (255) ,
    "09:00-10:00" VARCHAR (255) ,
    "10:00-11:00" VARCHAR (255) ,
    "11:00-12:00" VARCHAR (255) ,
    "12:00-13:00" VARCHAR (255) ,
```

```
"13:00-14:00" VARCHAR (255) ,
"14:00-15:00" VARCHAR (255) ,
"15:00-16:00" VARCHAR (255) ,
"16:00-17:00" VARCHAR (255) ,
"17:00-18:00" VARCHAR (255) ,
"18:00-19:00" VARCHAR (255) ,
"19:00-20:00" VARCHAR (255) ,
"20:00-21:00" VARCHAR (255) ,
"22:00-21:00" VARCHAR (255) ,
"22:00-23:00" VARCHAR (255) ,
"23:00-00:00" VARCHAR (255) ,
BBM_Horario_ID_Horario CHAR (8) NOT NULL
)
GO

CREATE TABLE BBM_Horario_Nocturno
(
    Numero_Camas INTEGER ,
    BBM_Horario_ID_Horario CHAR (8) NOT NULL
)
GO

CREATE TABLE BBM_IT
(
    BBM_Otros_Departamentos_BBM_Emppleado_BBM_Emppleado_ID INTEGER NOT
NULL ,
    Descripcion_funciones VARCHAR (255)
)
GO

ALTER TABLE BBM_IT ADD CONSTRAINT BBM_IT_PK PRIMARY KEY CLUSTERED
(BBM_Otros_Departamentos_BBM_Emppleado_BBM_Emppleado_ID)
    WITH (
        ALLOW_PAGE_LOCKS = ON ,
        ALLOW_ROW_LOCKS = ON )
GO

CREATE TABLE BBM_Logopeda
(
    BBM_Especialista_BBM_Emppleado_BBM_Emppleado_ID INTEGER NOT NULL ,
    Descripcion_funciones VARCHAR
)
GO

ALTER TABLE BBM_Logopeda ADD CONSTRAINT BBM_Logopeda_PK PRIMARY KEY
CLUSTERED (BBM_Especialista_BBM_Emppleado_BBM_Emppleado_ID)
    WITH (
        ALLOW_PAGE_LOCKS = ON ,
        ALLOW_ROW_LOCKS = ON )
GO

CREATE TABLE BBM_Mantenimiento
(
    BBM_Otros_Departamentos_BBM_Emppleado_BBM_Emppleado_ID INTEGER NOT
NULL ,
    Descripcion_funciones VARCHAR (255)
)
```

```
GO

ALTER TABLE BBM_Mantenimiento ADD CONSTRAINT BBM_Mantenimiento_PK PRIMARY KEY CLUSTERED (BBM_Otros_Departamentos_BBM_Empelado_BBM_Empelado_ID)
    WITH (
        ALLOW_PAGE_LOCKS = ON ,
        ALLOW_ROW_LOCKS = ON )
GO

CREATE TABLE BBM_Medicacion
(
    ID_Medicacion CHAR (12) NOT NULL ,
    Nombre_Medicamento VARCHAR (25) NOT NULL ,
    Codigo_EAN INTEGER NOT NULL ,
    Fabricante VARCHAR (50) ,
    Posologia VARCHAR (255) ,
    Descripcion VARCHAR (255) ,
    Observaciones VARCHAR (255)
)
GO

ALTER TABLE BBM_Medicacion ADD CONSTRAINT BBM_Medicacion_PK PRIMARY KEY CLUSTERED (ID_Medicacion)
    WITH (
        ALLOW_PAGE_LOCKS = ON ,
        ALLOW_ROW_LOCKS = ON )
GO

CREATE TABLE BBM_Medicacion_Paciente
(
    BBM_Medicacion_ID_Medicacion CHAR (12) NOT NULL ,
    BBM_Expediente_ID_Expediente VARCHAR (10) NOT NULL ,
    Toma_1 DATETIME ,
    Toma_2 DATETIME ,
    Toma_3 DATETIME ,
    Toma_4 DATETIME ,
    Toma_5 DATETIME
)
GO

ALTER TABLE BBM_Medicacion_Paciente ADD CONSTRAINT BBM_Medicacion_Paciente_PK PRIMARY KEY CLUSTERED (BBM_Medicacion_ID_Medicacion, BBM_Expediente_ID_Expediente)
    WITH (
        ALLOW_PAGE_LOCKS = ON ,
        ALLOW_ROW_LOCKS = ON )
GO

CREATE TABLE BBM_Metodo_Doman
(
    BBM_Otras_Terapias_BBM_Tratamientos_BBM_Tratamientos_ID INTEGER NOT NULL ,
    Descripcion VARCHAR (255) NOT NULL
)
GO
```

```
ALTER TABLE BBM_Metodo_Doman ADD CONSTRAINT BBM_Metodo_Doman_PK PRIMARY
KEY CLUSTERED (BBM_Otras_Terapias_BBM_Tratamientos_BBM_Tratamientos_ID)
    WITH (
        ALLOW_PAGE_LOCKS = ON ,
        ALLOW_ROW_LOCKS = ON )
GO

CREATE TABLE BBM_Otras_Terapias
(
    BBM_Tratamientos_BBM_Tratamientos_ID INTEGER NOT NULL ,
    OTT_ID CHAR (9) NOT NULL ,
    BBM_Tratamientos_BBM_Tratamientos_ID2 INTEGER NOT NULL
)
GO

ALTER TABLE BBM_Otras_Terapias ADD CONSTRAINT BBM_Otras_Terapias_PK
PRIMARY KEY CLUSTERED (BBM_Tratamientos_BBM_Tratamientos_ID)
    WITH (
        ALLOW_PAGE_LOCKS = ON ,
        ALLOW_ROW_LOCKS = ON )
GO

CREATE TABLE BBM_Otros_Departamentos
(
    BBM_Emppleado_BBM_Emppleado_ID INTEGER NOT NULL ,
    OD_ID CHAR (5) NOT NULL
)
GO

ALTER TABLE BBM_Otros_Departamentos ADD CONSTRAINT
BBM_Otros_Departamentos_PK PRIMARY KEY CLUSTERED
(BBM_Emppleado_BBM_Emppleado_ID)
    WITH (
        ALLOW_PAGE_LOCKS = ON ,
        ALLOW_ROW_LOCKS = ON )
GO
ALTER TABLE BBM_Otros_Departamentos ADD CONSTRAINT
BBM_Otros_Departamentos_OD_ID_UN UNIQUE NONCLUSTERED (OD_ID)
GO

CREATE TABLE BBM_Oxigencion_Hiperbarica
(
    BBM_Otras_Terapias_BBM_Tratamientos_BBM_Tratamientos_ID INTEGER NOT
NULL ,
    Descripcion VARCHAR (255)
)
GO

ALTER TABLE BBM_Oxigencion_Hiperbarica ADD CONSTRAINT
BBM_Oxigencion_Hiperbarica_PK PRIMARY KEY CLUSTERED
(BBM_Otras_Terapias_BBM_Tratamientos_BBM_Tratamientos_ID)
    WITH (
        ALLOW_PAGE_LOCKS = ON ,
        ALLOW_ROW_LOCKS = ON )
GO

CREATE TABLE BBM_Psicologo
```

```
(  
    BBM_Especialista_BBM_Emppleado_BBM_Emppleado_ID INTEGER NOT NULL ,  
    Descripcion_funciones VARCHAR (255)  
)  
GO  
  
ALTER TABLE BBM_Psicologo ADD CONSTRAINT BBM_Psicologo_PK PRIMARY KEY  
CLUSTERED (BBM_Especialista_BBM_Emppleado_BBM_Emppleado_ID)  
WITH (  
    ALLOW_PAGE_LOCKS = ON ,  
    ALLOW_ROW_LOCKS = ON )  
GO  
  
CREATE TABLE BBM_Sesion  
(  
    Sesion_ID CHAR (14) NOT NULL ,  
    Fecha_y_Hora_de_inicio DATETIME ,  
    Fecha_y_Hora_de_fin DATETIME ,  
    BBM_Horario_ID_Horario CHAR (8) NOT NULL ,  
    BBM_Espacio_ID_Espacio CHAR (10) NOT NULL ,  
    BBM_Equipamiento_ID_Equipamiento CHAR (10) NOT NULL ,  
    BBM_Emppleado_BBM_Emppleado_ID INTEGER NOT NULL ,  
    BBM_Actividades_BBM_Actividades_ID INTEGER NOT NULL  
)  
GO  
  
  
EXEC sp_addextendedproperty 'MS_Description' , 'Sesion + ddmmaa + orden  
ejemplo SES07122020010' , 'USER' , 'dbo' , 'TABLE' , 'BBM_Sesion'  
'COLUMN' , 'Session_ID'  
GO  
  
ALTER TABLE BBM_Sesion ADD CONSTRAINT BBM_Sesion_PK PRIMARY KEY CLUSTERED  
(Sesion_ID)  
WITH (  
    ALLOW_PAGE_LOCKS = ON ,  
    ALLOW_ROW_LOCKS = ON )  
GO  
  
CREATE TABLE BBM_Terapeuta_Ocupacional  
(  
    BBM_Especialista_BBM_Emppleado_BBM_Emppleado_ID INTEGER NOT NULL ,  
    Descripcion_funciones VARCHAR (255)  
)  
GO  
  
ALTER TABLE BBM_Terapeuta_Ocupacional ADD CONSTRAINT  
BBM_Terapeuta_Ocupacional_PK PRIMARY KEY CLUSTERED  
(BBM_Especialista_BBM_Emppleado_BBM_Emppleado_ID)  
WITH (  
    ALLOW_PAGE_LOCKS = ON ,  
    ALLOW_ROW_LOCKS = ON )  
GO  
  
CREATE TABLE BBM_Terapia_Logopeda
```

```
(  
    BBM_Tratamientos_BBM_Tratamientos_ID INTEGER NOT NULL ,  
    TL_ID CHAR (9) NOT NULL ,  
    Descripcion VARCHAR (255)  
)  
GO  
  
ALTER TABLE BBM_Terapia_Logopeda ADD CONSTRAINT BBM_Terapia_Logopeda_PK  
PRIMARY KEY CLUSTERED (BBM_Tratamientos_BBM_Tratamientos_ID)  
    WITH (  
        ALLOW_PAGE_LOCKS = ON ,  
        ALLOW_ROW_LOCKS = ON )  
GO  
  
CREATE TABLE BBM_Terapia_Ocupacional  
(  
    BBM_Tratamientos_BBM_Tratamientos_ID INTEGER NOT NULL ,  
    TO_ID CHAR (9) NOT NULL ,  
    Descripcion VARCHAR (255)  
)  
GO  
  
ALTER TABLE BBM_Terapia_Ocupacional ADD CONSTRAINT  
BBM_Terapia_Ocupacional_PK PRIMARY KEY CLUSTERED  
(BBM_Tratamientos_BBM_Tratamientos_ID)  
    WITH (  
        ALLOW_PAGE_LOCKS = ON ,  
        ALLOW_ROW_LOCKS = ON )  
GO  
  
CREATE TABLE BBM_Terapia_Recreativa  
(  
    BBM_Tratamientos_BBM_Tratamientos_ID INTEGER NOT NULL ,  
    TR_ID CHAR (9) NOT NULL ,  
    Descripcion VARCHAR (255) ,  
    AUX_ID CHAR NOT NULL  
)  
GO  
  
ALTER TABLE BBM_Terapia_Recreativa ADD CONSTRAINT  
BBM_Terapia_Recreativa_PK PRIMARY KEY CLUSTERED  
(BBM_Tratamientos_BBM_Tratamientos_ID)  
    WITH (  
        ALLOW_PAGE_LOCKS = ON ,  
        ALLOW_ROW_LOCKS = ON )  
GO  
  
CREATE TABLE BBM_Titulado  
(  
    BBM_Auxiliar_BBM_Empleado_BBM_Empleado_ID INTEGER NOT NULL ,  
    Descripcion_funciones VARCHAR (255)  
)  
GO  
  
ALTER TABLE BBM_Titulado ADD CONSTRAINT BBM_Titulado_PK PRIMARY KEY  
CLUSTERED (BBM_Auxiliar_BBM_Empleado_BBM_Empleado_ID)  
    WITH (
```

```
    ALLOW_PAGE_LOCKS = ON ,
    ALLOW_ROW_LOCKS = ON )
GO

CREATE TABLE BBM_Tratamientos
(
    BBM_Tratamientos_ID INTEGER NOT NULL IDENTITY NOT FOR REPLICATION ,
    Tratamiento_ID CHAR (4) NOT NULL ,
    BBM_Auxiliar_BBM_Emppleado_BBM_Emppleado_ID INTEGER NOT NULL ,
    BBM_Especialista_BBM_Emppleado_BBM_Emppleado_ID INTEGER NOT NULL
)
GO

ALTER TABLE BBM_Tratamientos ADD CONSTRAINT BBM_Tratamientos_PK PRIMARY
KEY CLUSTERED (BBM_Tratamientos_ID)
    WITH (
        ALLOW_PAGE_LOCKS = ON ,
        ALLOW_ROW_LOCKS = ON )
GO

CREATE TABLE BBM_Tutor
(
    Tutor_ID VARCHAR (10) NOT NULL ,
    Nombre_1 VARCHAR (20) ,
    Nombre_2 VARCHAR (20) ,
    Apellido_1 VARCHAR (20) ,
    Apellido_2 VARCHAR (20) ,
    DNI VARCHAR (9) NOT NULL ,
    Otros_Detalles VARCHAR (255)
)
GO
```

```
EXEC sp_addextendedproperty 'MS_Description' , 'Codigo compuesto por TUT
+ ID del Usuario (Apellido + Apellido + Nombre + Numero)
ejemplo -> TUTBBM0001' , 'USER' , 'dbo' , 'TABLE' , 'BBM_Tutor' ,
'COLUMN' , 'Tutor_ID'
GO
```

```
EXEC sp_addextendedproperty 'MS_Description' , 'Para decir cosas que
pueden ir adicionales' , 'USER' , 'dbo' , 'TABLE' , 'BBM_Tutor' ,
'COLUMN' , 'Otros_Detalles'
GO
```

```
ALTER TABLE BBM_Tutor ADD CONSTRAINT BBM_Tutor_PK PRIMARY KEY CLUSTERED
(Tutor_ID)
    WITH (
        ALLOW_PAGE_LOCKS = ON ,
        ALLOW_ROW_LOCKS = ON )
GO
```

```
CREATE TABLE BBM_Usuario
(
    Usuario_ID VARCHAR (10) NOT NULL ,
```

```
Nombre_1 VARCHAR (20) ,
Nombre_2 VARCHAR (20) ,
Apellido_1 VARCHAR (20) ,
Apellido_2 VARCHAR (20) ,
DNI VARCHAR (9) NOT NULL ,
Otros_Detalles VARCHAR (255)
)
GO

EXEC sp_addextendedproperty 'MS_Description' , 'Codigo compuesto por PACX
+ ID del Usuario (Apellido + Apellido + Nombre + Numero)
ejemplo -> PACBBM0001' , 'USER' , 'dbo' , 'TABLE' , 'BBM_Usuario' ,
'COLUMN' , 'Usuario_ID'
GO

EXEC sp_addextendedproperty 'MS_Description' , 'Por si hay algo que allar a
mayores, como puede ser un segundo contacto' , 'USER' , 'dbo' , 'TABLE' ,
'BBM_Usuario' , 'COLUMN' , 'Otros_Detalles'
GO

ALTER TABLE BBM_Usuario ADD CONSTRAINT BBM_Usuario_PK PRIMARY KEY
CLUSTERED (Usuario_ID)
    WITH (
        ALLOW_PAGE_LOCKS = ON ,
        ALLOW_ROW_LOCKS = ON )
GO

CREATE TABLE BBM_Voluntario
(
    BBM_Auxiliar_BBM_Empleado_BBM_Empleado_ID INTEGER NOT NULL ,
    Descripcion_funciones VARCHAR (255)
)
GO

ALTER TABLE BBM_Voluntario ADD CONSTRAINT BBM_Voluntario_PK PRIMARY KEY
CLUSTERED (BBM_Auxiliar_BBM_Empleado_BBM_Empleado_ID)
    WITH (
        ALLOW_PAGE_LOCKS = ON ,
        ALLOW_ROW_LOCKS = ON )
GO

ALTER TABLE BBM_Actividades
    ADD CONSTRAINT BBM_Actividades_BBM_Auxiliar_FK FOREIGN KEY
    (
        BBM_Auxiliar_BBM_Empleado_BBM_Empleado_ID
    )
    REFERENCES BBM_Auxiliar
    (
        BBM_Empleado_BBM_Empleado_ID
    )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
GO
```

```
ALTER TABLE BBM_Asistente_Social
ADD CONSTRAINT BBM_Asistente_Social_BBMM_Especialista_FK FOREIGN KEY
(
    BBM_Especialista_BBMM_Empleado_BBMM_Empleado_ID
)
REFERENCES BBM_Especialista
(
    BBM_Empleado_BBMM_Empleado_ID
)
ON DELETE NO ACTION
ON UPDATE NO ACTION
GO

ALTER TABLE BBM_Auxiliar
ADD CONSTRAINT BBM_Auxiliar_BBMM_Empleado_FK FOREIGN KEY
(
    BBM_Empleado_BBMM_Empleado_ID
)
REFERENCES BBM_Empleado
(
    BBM_Empleado_ID
)
ON DELETE NO ACTION
ON UPDATE NO ACTION
GO

ALTER TABLE BBM_Cocina
ADD CONSTRAINT BBM_Cocina_BBMM_Otros_Departamentos_FK FOREIGN KEY
(
    BBM_Otros_Departamentos_BBMM_Empleado_BBMM_Empleado_ID
)
REFERENCES BBM_Otros_Departamentos
(
    BBM_Empleado_BBMM_Empleado_ID
)
ON DELETE NO ACTION
ON UPDATE NO ACTION
GO

ALTER TABLE BBM_Comida_Paciente
ADD CONSTRAINT BBM_Comida_Paciente_BBMM_Comidas_FK FOREIGN KEY
(
    BBM_Comidas_ID_Menu
)
REFERENCES BBM_Comidas
(
    ID_Menu
)
ON DELETE NO ACTION
ON UPDATE NO ACTION
GO

ALTER TABLE BBM_Comida_Paciente
ADD CONSTRAINT BBM_Comida_Paciente_BBMM_Expediente_FK FOREIGN KEY
(
    BBM_Expediente_ID_Expediente
```

```
)  
REFERENCES BBM_Expediente  
(  
    ID_Expediente  
)  
ON DELETE NO ACTION  
ON UPDATE NO ACTION  
GO  
  
ALTER TABLE BBM_Comidas  
ADD CONSTRAINT BBM_Comidas_BBM_Otros_Departamentos_FK FOREIGN KEY  
(  
    BBM_Otros_Departamentos_OD_ID  
)  
REFERENCES BBM_Otros_Departamentos  
(  
    OD_ID  
)  
ON DELETE NO ACTION  
ON UPDATE NO ACTION  
GO  
  
ALTER TABLE BBM_Contacto_Paciente  
ADD CONSTRAINT BBM_Contacto_Paciente_BBM_Tutor_FK FOREIGN KEY  
(  
    BBM_Tutor_Tutor_ID  
)  
REFERENCES BBM_Tutor  
(  
    Tutor_ID  
)  
ON DELETE NO ACTION  
ON UPDATE NO ACTION  
GO  
  
ALTER TABLE BBM_Contacto_Paciente  
ADD CONSTRAINT BBM_Contacto_Paciente_BBM_Usuario_FK FOREIGN KEY  
(  
    BBM_Usuario_Usuario_ID  
)  
REFERENCES BBM_Usuario  
(  
    Usuario_ID  
)  
ON DELETE NO ACTION  
ON UPDATE NO ACTION  
GO  
  
ALTER TABLE BBM_Direccion  
ADD CONSTRAINT BBM_Direccion_BBM_Otros_Departamentos_FK FOREIGN KEY  
(  
    BBM_Otros_Departamentos_BBM_Empleado_BBM_Empleado_ID  
)  
REFERENCES BBM_Otros_Departamentos  
(  
    BBM_Empleado_BBM_Empleado_ID  
)
```

```
        ON DELETE NO ACTION
        ON UPDATE NO ACTION
GO

ALTER TABLE BBM_Enfermera
    ADD CONSTRAINT BBM_Enfermera_BBMM_Especialista_FK FOREIGN KEY
    (
        BBM_Especialista_BBMM_Emppleado_BBMM_Emppleado_ID
    )
    REFERENCES BBM_Especialista
    (
        BBM_Emppleado_BBMM_Emppleado_ID
    )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
GO

ALTER TABLE BBM_Especialista
    ADD CONSTRAINT BBM_Especialista_BBMM_Emppleado_FK FOREIGN KEY
    (
        BBM_Emppleado_BBMM_Emppleado_ID
    )
    REFERENCES BBM_Emppleado
    (
        BBM_Emppleado_ID
    )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
GO

ALTER TABLE BBM_Estimulacion_Electrica
    ADD CONSTRAINT BBM_Estimulacion_Electrica_BBMM_Otras_Terapias_FK
FOREIGN KEY
    (
        BBM_Otras_Terapias_BBMM_Tratamientos_BBMM_Tratamientos_ID
    )
    REFERENCES BBM_Otras_Terapias
    (
        BBM_Tratamientos_BBMM_Tratamientos_ID
    )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
GO

ALTER TABLE BBM_Expediente
    ADD CONSTRAINT BBM_Expediente_BBMM_Enfermedad_Diagnosticada_FK FOREIGN
KEY
    (
        BBM_Enfermedad_Diagnosticada_Nombre_Enfermedad
    )
    REFERENCES BBM_Enfermedad_Diagnosticada
    (
        Nombre_Enfermedad
    )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
GO
```

```
ALTER TABLE BBM_Expediente
  ADD CONSTRAINT BBM_Expediente_BBM_Sesion_FK FOREIGN KEY
  (
    BBM_Sesion_Sesion_ID
  )
  REFERENCES BBM_Session
  (
    Session_ID
  )
  ON DELETE NO ACTION
  ON UPDATE NO ACTION
GO

ALTER TABLE BBM_Expediente
  ADD CONSTRAINT BBM_Expediente_BBM_Tratamientos_FK FOREIGN KEY
  (
    BBM_Tratamientos_BBM_Tratamientos_ID
  )
  REFERENCES BBM_Tratamientos
  (
    BBM_Tratamientos_ID
  )
  ON DELETE NO ACTION
  ON UPDATE NO ACTION
GO

ALTER TABLE BBM_Expediente
  ADD CONSTRAINT BBM_Expediente_BBM_Usuario_FK FOREIGN KEY
  (
    BBM_Usuario_Usuario_ID
  )
  REFERENCES BBM_Usuario
  (
    Usuario_ID
  )
  ON DELETE NO ACTION
  ON UPDATE NO ACTION
GO

ALTER TABLE BBM_Factura
  ADD CONSTRAINT BBM_Factura_BBM_Horario_FK FOREIGN KEY
  (
    BBM_Horario_ID_Horario
  )
  REFERENCES BBM_Horario
  (
    ID_Horario
  )
  ON DELETE NO ACTION
  ON UPDATE NO ACTION
GO

ALTER TABLE BBM_Factura
  ADD CONSTRAINT BBM_Factura_BBM_Tratamientos_FK FOREIGN KEY
  (
    BBM_Tratamientos_BBM_Tratamientos_ID
  )
```

```
)  
REFERENCES BBM_Tratamientos  
(  
    BBM_Tratamientos_ID  
)  
ON DELETE NO ACTION  
ON UPDATE NO ACTION  
GO  
  
ALTER TABLE BBM_Factura  
    ADD CONSTRAINT BBM_Factura_BBM_Tutor_FK FOREIGN KEY  
(  
    BBM_Tutor_Tutor_ID  
)  
REFERENCES BBM_Tutor  
(  
    Tutor_ID  
)  
ON DELETE NO ACTION  
ON UPDATE NO ACTION  
GO  
  
ALTER TABLE BBM_Fisioterapeutas  
    ADD CONSTRAINT BBM_Fisioterapeutas_BBM_Especialista_FK FOREIGN KEY  
(  
    BBM_Especialista_BBM_Emppleado_BBM_Emppleado_ID  
)  
REFERENCES BBM_Especialista  
(  
    BBM_Emppleado_BBM_Emppleado_ID  
)  
ON DELETE NO ACTION  
ON UPDATE NO ACTION  
GO  
  
ALTER TABLE BBM_Fisioterapia  
    ADD CONSTRAINT BBM_Fisioterapia_BBM_Tratamientos_FK FOREIGN KEY  
(  
    BBM_Tratamientos_BBM_Tratamientos_ID  
)  
REFERENCES BBM_Tratamientos  
(  
    BBM_Tratamientos_ID  
)  
ON DELETE NO ACTION  
ON UPDATE NO ACTION  
GO  
  
ALTER TABLE BBM_Horario_Diurno  
    ADD CONSTRAINT BBM_Horario_Diurno_BBM_Horario_FK FOREIGN KEY  
(  
    BBM_Horario_ID_Horario  
)  
REFERENCES BBM_Horario  
(  
    ID_Horario  
)
```

```
        ON DELETE NO ACTION
        ON UPDATE NO ACTION
GO

ALTER TABLE BBM_Horario_Fabricado
    ADD CONSTRAINT BBM_Horario_Fabricado_BBM_Horario_FK FOREIGN KEY
    (
        BBM_Horario_ID_Horario
    )
    REFERENCES BBM_Horario
    (
        ID_Horario
    )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
GO

ALTER TABLE BBM_Horario_Nocturno
    ADD CONSTRAINT BBM_Horario_Nocturno_BBM_Horario_FK FOREIGN KEY
    (
        BBM_Horario_ID_Horario
    )
    REFERENCES BBM_Horario
    (
        ID_Horario
    )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
GO

ALTER TABLE BBM_IT
    ADD CONSTRAINT BBM_IT_BBM_Otros_Departamentos_FK FOREIGN KEY
    (
        BBM_Otros_Departamentos_BBM_Empleado_BBM_Empleado_ID
    )
    REFERENCES BBM_Otros_Departamentos
    (
        BBM_Empleado_BBM_Empleado_ID
    )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
GO

ALTER TABLE BBM_Logopeda
    ADD CONSTRAINT BBM_Logopeda_BBM_Especialista_FK FOREIGN KEY
    (
        BBM_Especialista_BBM_Empleado_BBM_Empleado_ID
    )
    REFERENCES BBM_Especialista
    (
        BBM_Empleado_BBM_Empleado_ID
    )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
GO

ALTER TABLE BBM_Mantenimiento
```

```
ADD CONSTRAINT BBM_Mantenimiento_BBM_Otros_Departamentos_FK FOREIGN  
KEY  
(  
    BBM_Otros_Departamentos_BBM_Empleado_BBM_Empleado_ID  
)  
REFERENCES BBM_Otros_Departamentos  
(  
    BBM_Empleado_BBM_Empleado_ID  
)  
ON DELETE NO ACTION  
ON UPDATE NO ACTION  
GO  
  
ALTER TABLE BBM_Medicacion_Paciente  
ADD CONSTRAINT BBM_Medicacion_Paciente_BBM_Expediente_FK FOREIGN KEY  
(  
    BBM_Expediente_ID_Expediente  
)  
REFERENCES BBM_Expediente  
(  
    ID_Expediente  
)  
ON DELETE NO ACTION  
ON UPDATE NO ACTION  
GO  
  
ALTER TABLE BBM_Medicacion_Paciente  
ADD CONSTRAINT BBM_Medicacion_Paciente_BBM_Medicacion_FK FOREIGN KEY  
(  
    BBM_Medicacion_ID_Medicacion  
)  
REFERENCES BBM_Medicacion  
(  
    ID_Medicacion  
)  
ON DELETE NO ACTION  
ON UPDATE NO ACTION  
GO  
  
ALTER TABLE BBM_Metodo_Doman  
ADD CONSTRAINT BBM_Metodo_Doman_BBM_Otras_Terapias_FK FOREIGN KEY  
(  
    BBM_Otras_Terapias_BBM_Tratamientos_BBM_Tratamientos_ID  
)  
REFERENCES BBM_Otras_Terapias  
(  
    BBM_Tratamientos_BBM_Tratamientos_ID  
)  
ON DELETE NO ACTION  
ON UPDATE NO ACTION  
GO  
  
ALTER TABLE BBM_Otras_Terapias  
ADD CONSTRAINT BBM_Otras_Terapias_BBM_Tratamientos_FK FOREIGN KEY  
(  
    BBM_Tratamientos_BBM_Tratamientos_ID  
)
```

```
REFERENCES BBM_Tratamientos
(
    BBM_Tratamientos_ID
)
ON DELETE NO ACTION
ON UPDATE NO ACTION
GO

ALTER TABLE BBM_Otras_Terapias
ADD CONSTRAINT BBM_Otras_Terapias_BBM_Tratamientos_FKv2 FOREIGN KEY
(
    BBM_Tratamientos_BBM_Tratamientos_ID2
)
REFERENCES BBM_Tratamientos
(
    BBM_Tratamientos_ID
)
ON DELETE NO ACTION
ON UPDATE NO ACTION
GO

ALTER TABLE BBM_Otros_Departamentos
ADD CONSTRAINT BBM_Otros_Departamentos_BBM_Empleado_FK FOREIGN KEY
(
    BBM_Empleado_BBM_Empleado_ID
)
REFERENCES BBM_Empleado
(
    BBM_Empleado_ID
)
ON DELETE NO ACTION
ON UPDATE NO ACTION
GO

ALTER TABLE BBM_Oxigencion_Hiperbarica
ADD CONSTRAINT BBM_Oxigencion_Hiperbarica_BBM_Otras_Terapias_FK
FOREIGN KEY
(
    BBM_Otras_Terapias_BBM_Tratamientos_BBM_Tratamientos_ID
)
REFERENCES BBM_Otras_Terapias
(
    BBM_Tratamientos_BBM_Tratamientos_ID
)
ON DELETE NO ACTION
ON UPDATE NO ACTION
GO

ALTER TABLE BBM_Psicologo
ADD CONSTRAINT BBM_Psicologo_BBM_Especialista_FK FOREIGN KEY
(
    BBM_Especialista_BBM_Empleado_BBM_Empleado_ID
)
REFERENCES BBM_Especialista
(
    BBM_Empleado_BBM_Empleado_ID
)
```

```
        ON DELETE NO ACTION
        ON UPDATE NO ACTION
GO

ALTER TABLE BBM_Sesion
    ADD CONSTRAINT BBM_Sesion_BBMMActividades_FK FOREIGN KEY
    (
        BBM_Actividades_BBMMActividades_ID
    )
    REFERENCES BBM_Actividades
    (
        BBM_Actividades_ID
    )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
GO

ALTER TABLE BBM_Sesion
    ADD CONSTRAINT BBM_Sesion_BBMEmppleado_FK FOREIGN KEY
    (
        BBM_Emppleado_BBMEmppleado_ID
    )
    REFERENCES BBM_Emppleado
    (
        BBM_Emppleado_ID
    )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
GO

ALTER TABLE BBM_Sesion
    ADD CONSTRAINT BBM_Sesion_BBMEquipamiento_FK FOREIGN KEY
    (
        BBM_Equipamiento_ID_Equipamiento
    )
    REFERENCES BBM_Equipamiento
    (
        ID_Equipamiento
    )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
GO

ALTER TABLE BBM_Sesion
    ADD CONSTRAINT BBM_Sesion_BBMEspacio_FK FOREIGN KEY
    (
        BBM_Espacio_ID_Espacio
    )
    REFERENCES BBM_Espacio
    (
        ID_Espacio
    )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
GO

ALTER TABLE BBM_Sesion
```

```
ADD CONSTRAINT BBM_Sesion_BBM_Horario_FK FOREIGN KEY
(
    BBM_Horario_ID_Horario
)
REFERENCES BBM_Horario
(
    ID_Horario
)
ON DELETE NO ACTION
ON UPDATE NO ACTION
GO

ALTER TABLE BBM_Terapeuta_Ocupacional
ADD CONSTRAINT BBM_Terapeuta_Ocupacional_BBM_Especialista_FK FOREIGN
KEY
(
    BBM_Especialista_BBM_Empleado_BBM_Empleado_ID
)
REFERENCES BBM_Especialista
(
    BBM_Empleado_BBM_Empleado_ID
)
ON DELETE NO ACTION
ON UPDATE NO ACTION
GO

ALTER TABLE BBM_Terapia_Logopeda
ADD CONSTRAINT BBM_Terapia_Logopeda_BBM_Tratamientos_FK FOREIGN KEY
(
    BBM_Tratamientos_BBM_Tratamientos_ID
)
REFERENCES BBM_Tratamientos
(
    BBM_Tratamientos_ID
)
ON DELETE NO ACTION
ON UPDATE NO ACTION
GO

ALTER TABLE BBM_Terapia_Ocupacional
ADD CONSTRAINT BBM_Terapia_Ocupacional_BBM_Tratamientos_FK FOREIGN
KEY
(
    BBM_Tratamientos_BBM_Tratamientos_ID
)
REFERENCES BBM_Tratamientos
(
    BBM_Tratamientos_ID
)
ON DELETE NO ACTION
ON UPDATE NO ACTION
GO

ALTER TABLE BBM_Terapia_Recreativa
ADD CONSTRAINT BBM_Terapia_Recreativa_BBM_Tratamientos_FK FOREIGN KEY
(
    BBM_Tratamientos_BBM_Tratamientos_ID
)
```

```
)  
REFERENCES BBM_Tratamientos  
(  
    BBM_Tratamientos_ID  
)  
ON DELETE NO ACTION  
ON UPDATE NO ACTION  
GO  
  
ALTER TABLE BBM_Titulado  
ADD CONSTRAINT BBM_Titulado_BBMM_Auxiliar_FK FOREIGN KEY  
(  
    BBM_Auxiliar_BBMM_Empleado_BBMM_Empleado_ID  
)  
REFERENCES BBM_Auxiliar  
(  
    BBM_Empleado_BBMM_Empleado_ID  
)  
ON DELETE NO ACTION  
ON UPDATE NO ACTION  
GO  
  
ALTER TABLE BBM_Tratamientos  
ADD CONSTRAINT BBM_Tratamientos_BBMM_Auxiliar_FK FOREIGN KEY  
(  
    BBM_Auxiliar_BBMM_Empleado_BBMM_Empleado_ID  
)  
REFERENCES BBM_Auxiliar  
(  
    BBM_Empleado_BBMM_Empleado_ID  
)  
ON DELETE NO ACTION  
ON UPDATE NO ACTION  
GO  
  
ALTER TABLE BBM_Tratamientos  
ADD CONSTRAINT BBM_Tratamientos_BBMM_Especialista_FK FOREIGN KEY  
(  
    BBM_Especialista_BBMM_Empleado_BBMM_Empleado_ID  
)  
REFERENCES BBM_Especialista  
(  
    BBM_Empleado_BBMM_Empleado_ID  
)  
ON DELETE NO ACTION  
ON UPDATE NO ACTION  
GO  
  
ALTER TABLE BBM_Voluntario  
ADD CONSTRAINT BBM_Voluntario_BBMM_Auxiliar_FK FOREIGN KEY  
(  
    BBM_Auxiliar_BBMM_Empleado_BBMM_Empleado_ID  
)  
REFERENCES BBM_Auxiliar  
(  
    BBM_Empleado_BBMM_Empleado_ID  
)
```

```
ON DELETE NO ACTION  
ON UPDATE NO ACTION  
GO
```

```
-- Oracle SQL Developer Data Modeler Summary Report:  
--  
-- CREATE TABLE 43  
-- CREATE INDEX 0  
-- ALTER TABLE 90  
-- CREATE VIEW 0  
-- ALTER VIEW 0  
-- CREATE PACKAGE 0  
-- CREATE PACKAGE BODY 0  
-- CREATE PROCEDURE 0  
-- CREATE FUNCTION 0  
-- CREATE TRIGGER 0  
-- ALTER TRIGGER 0  
-- CREATE DATABASE 0  
-- CREATE DEFAULT 0  
-- CREATE INDEX ON VIEW 0  
-- CREATE ROLLBACK SEGMENT 0  
-- CREATE ROLE 0  
-- CREATE RULE 0  
-- CREATE SCHEMA 0  
-- CREATE SEQUENCE 0  
-- CREATE PARTITION FUNCTION 0  
-- CREATE PARTITION SCHEME 0  
--  
-- DROP DATABASE 0  
--  
-- ERRORS 0  
-- WARNINGS 0
```

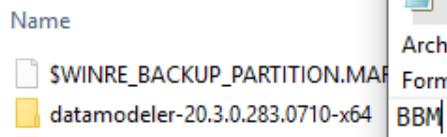
SQL Data Modeler a SQL Server

Antes de explicar el paso de Data Modeler a SQL Server, puesto que en anteriores secciones he dado pinceladas dentro del programa, pero no he explicado qué es **SQL**

Data Modeler: Oracle Data Modeler es una herramienta gráfica gratuita que mejora la productividad y simplifica las tareas de modelado de datos. Al utilizar Oracle SQL Developer Data Modeler, los usuarios pueden crear, navegar y editar modelos lógicos, relacionales, físicos, multidimensionales y de tipo de datos. Data Modeler proporciona capacidades de ingeniería directa e inversa y admite el desarrollo colaborativo a través del control de código fuente integrado. Data Modeler se puede utilizar tanto en entornos tradicionales como en la nube. Es fácil de instalar y no tiene coste alguno. de sus características: Se descarga en <https://www.oracle.com/tools/downloads/sql-data-modeler-downloads.html> y entre sus funciones se encuentra:

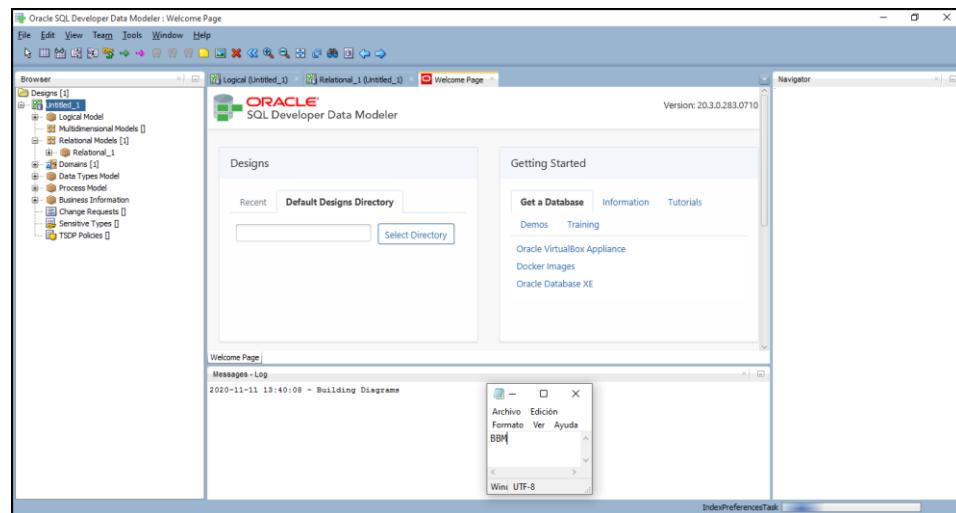
- Los modelos se almacenan en el sistema de ficheros, bajo una estructura de directorios (por lo que son fáciles de transportar, archivar, etc.)
- Puede trabajar con cualquier base de datos, no está restringido a Oracle
- Permite realizar ingeniería inversa
- Dispone de los siguientes niveles de diseño: lógico, relacional y físico, con herramientas de generación automática en ambos sentidos.
- Cada modelo puede tener diferentes implementaciones físicas (en diferentes tecnologías)
- Permite definir dominios de tipos de datos
- Compara diferencias entre modelos
- Soporte para código almacenado, vistas materializadas, etc. (no sólo tablas y vistas)
- Versionado de objetos
- Herramienta de diseño visual y rica en herramientas (colores personalizables, deshacer, búsqueda de objetos, etc.)
- Múltiples opciones en la generación del DDL

is PC > Local Disk (C:) >

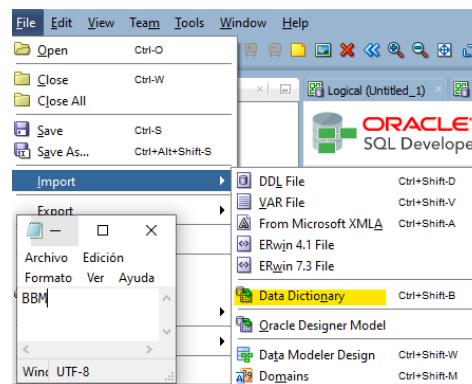


Su instalación es muy sencilla. Después de descargarlo (es una versión portable), lo descomprimimos y pegamos en C:/.

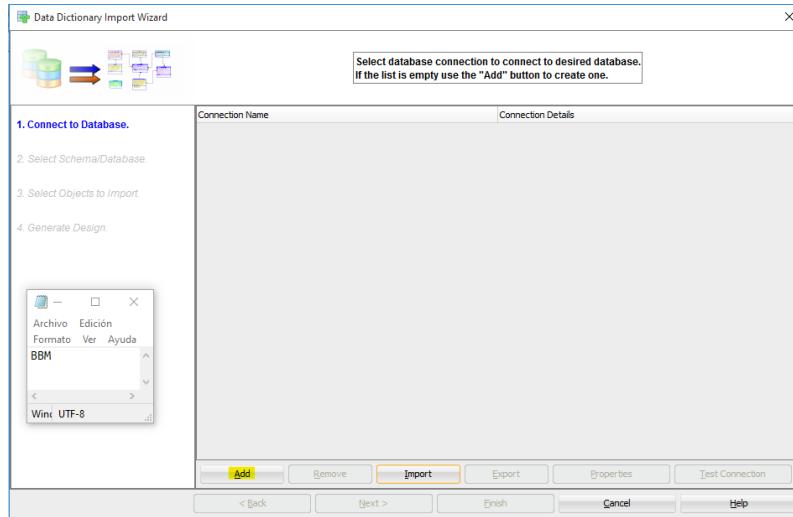
Una vez hecho esto, lo iniciamos como administrador. Un dato curioso es que su entorno es prácticamente igual al de *SQL Developer*.



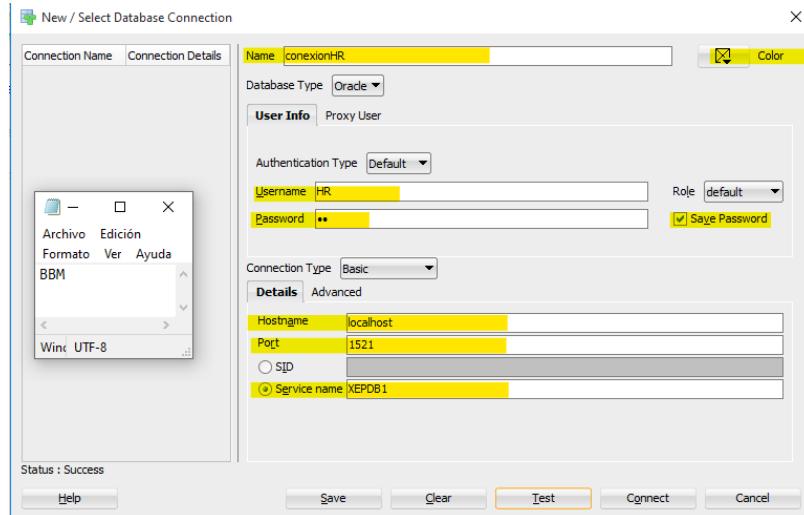
La conexión se hace de la siguiente manera → Hacemos clic en **File>Import>Data Dictionary**.



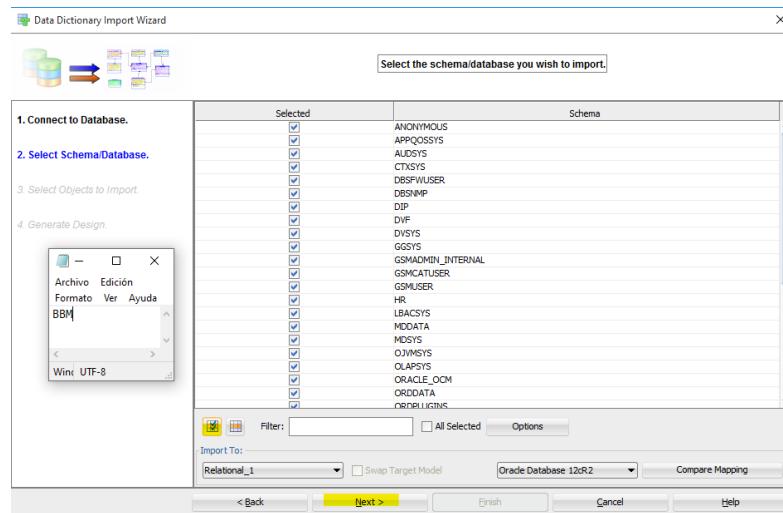
Nos va a salir un asistente, en el que en la primera ventana hacemos clic en **Add**.



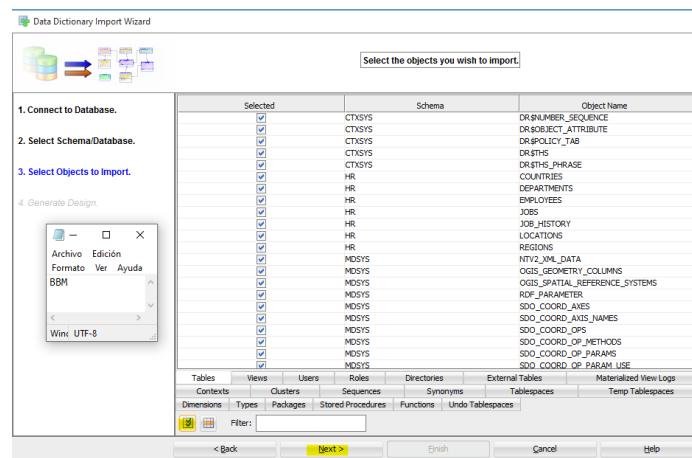
Se abre una sub-ventana que es exactamente igual que las conexiones que tenemos en *SQL Developer* (en este caso dejo la conexión HR como ejemplo).



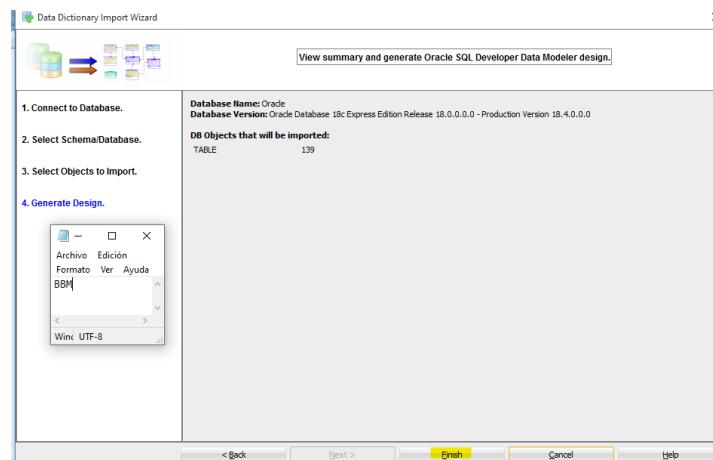
Una vez le damos a **Next**, hay que elegir los schemas que nos interesan. En caso de querer todos, pulsamos este botón .



Repetimos el proceso pulsando el botón o marcando lo que nos interesa para seleccionar los objetos que queremos importar:

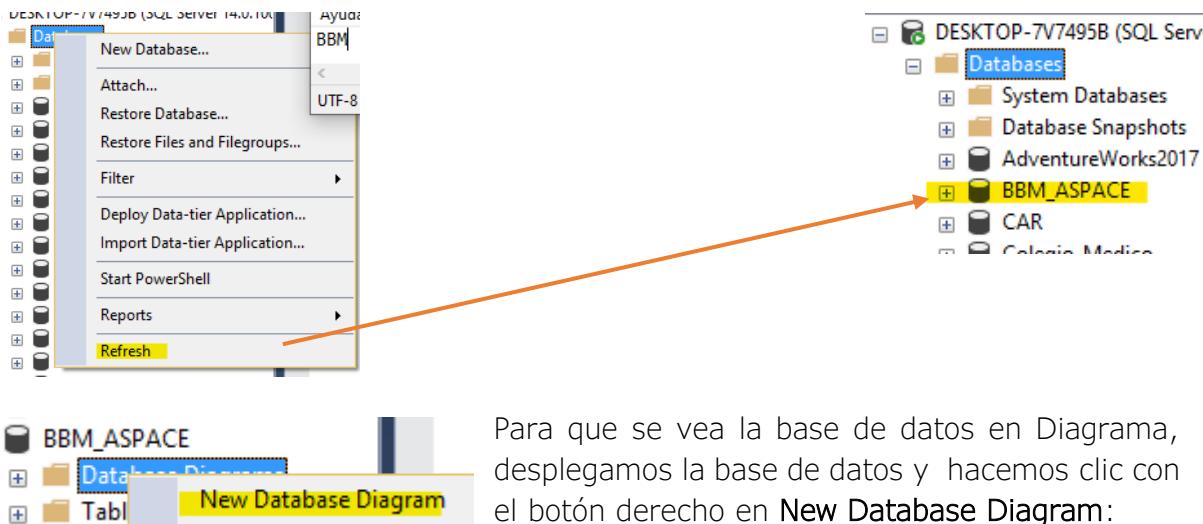


Por último, ya tenemos la generación del diseño. Pulsamos *Finish* y daríamos por concluida la importación de un usuario:



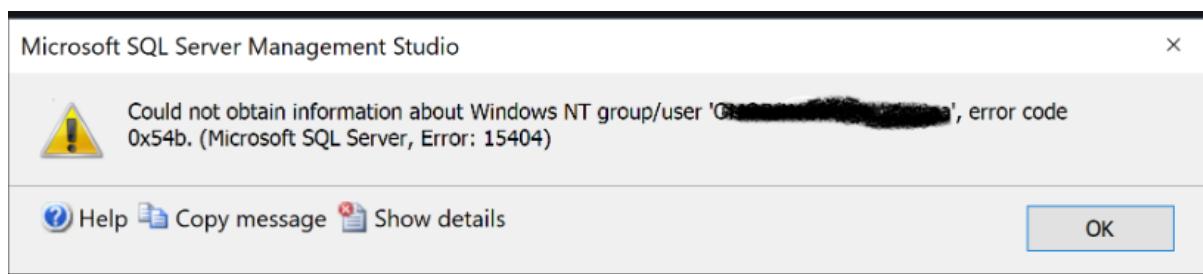
Convertir el modelo Lógico-Relacional a SQL Server

Una vez exportada la base de datos del modelo lógico-relacional, abrimos el SSMS y le damos al botón **New Query** y pegamos el script que generamos en el data modeler. Cuando lo tenemos bien revisado, le damos al botón **Execute**. Una vez acabado le damos a Refresh.



Para que se vea la base de datos en Diagrama, desplegamos la base de datos y hacemos clic con el botón derecho en **New Database Diagram**:

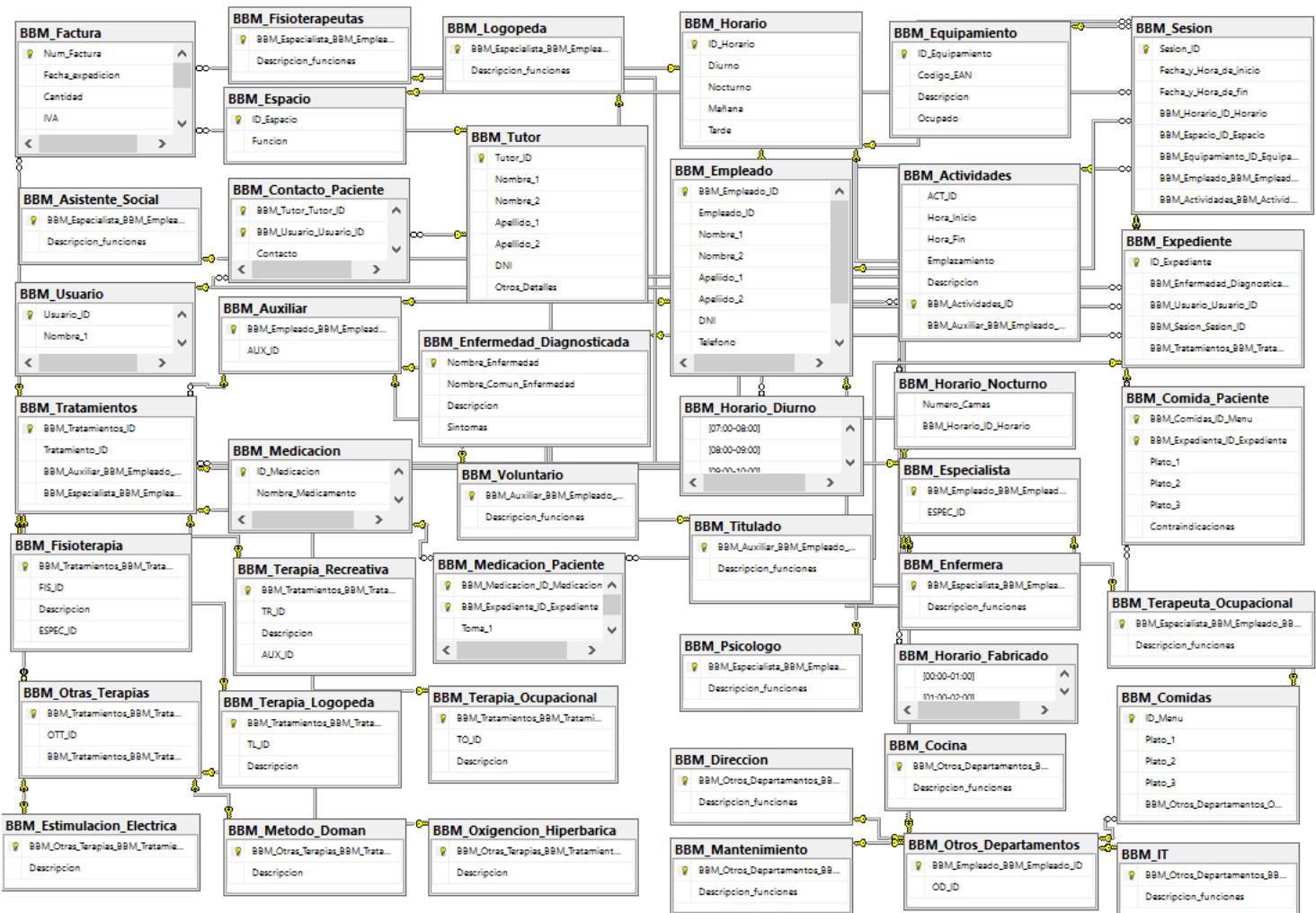
NOTA: Debido a un problema de permisos me saltaba el siguiente error:



Para solventarlo usé estos comandos:

```
USE BBM_ASPACE
GO
ALTER DATABASE BBM_ASPACE set TRUSTWORTHY ON;
GO
EXEC dbo.sp_changedbowner @loginame = N'sa', @map = false
GO
sp_configure 'show advanced options', 1;
GO
RECONFIGURE;
GO
sp_configure 'clr enabled', 1;
GO
RECONFIGURE;
GO
```

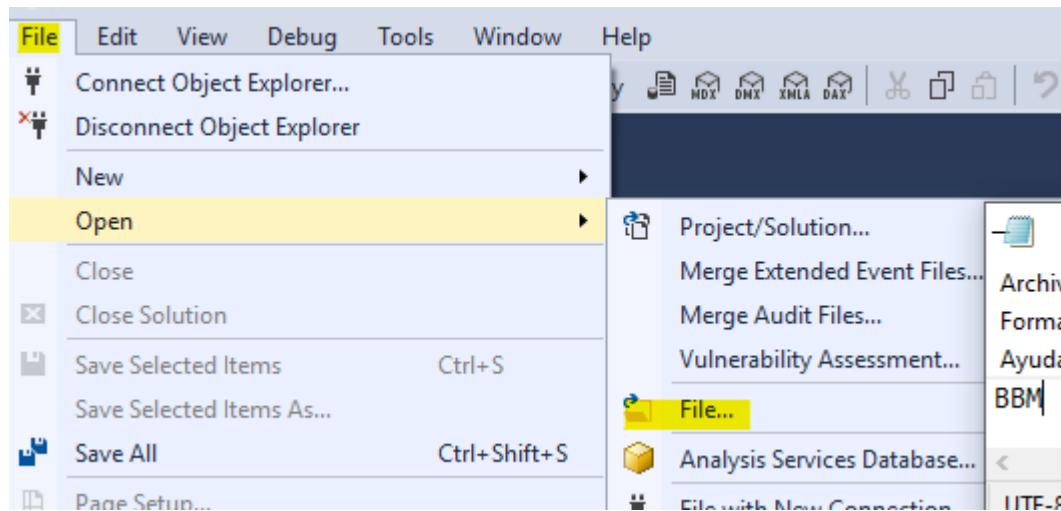
Dejo el diagrama a continuación:



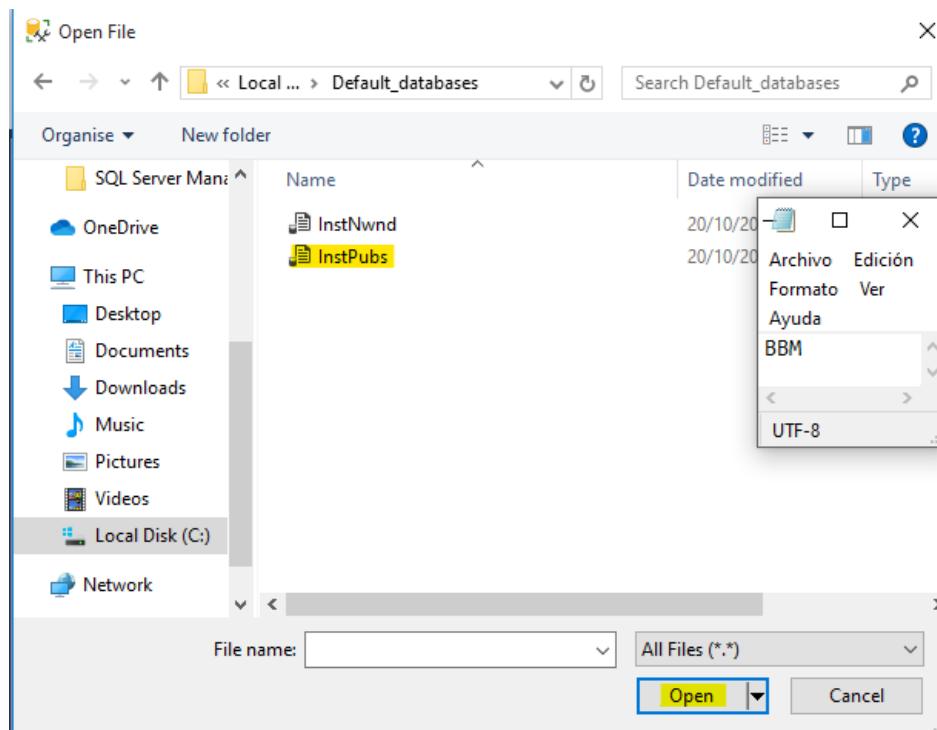
Instalar Bases de Datos de ejemplo en SSMS

Pubs desde Script

Hacemos clic en Archivo>Abrir>Archivo...



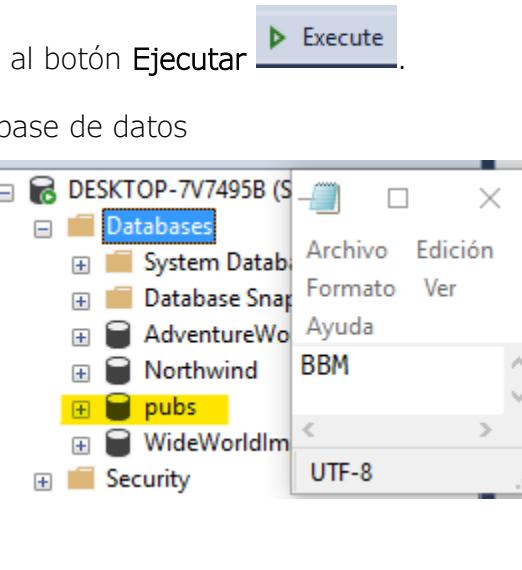
Seleccionamos el script .sql que nos interesa (en este caso **InstPubs**):



Como apunte, tenemos que comentar un par de líneas de la base de datos que vamos a instalar porque SQL no las reconoce:

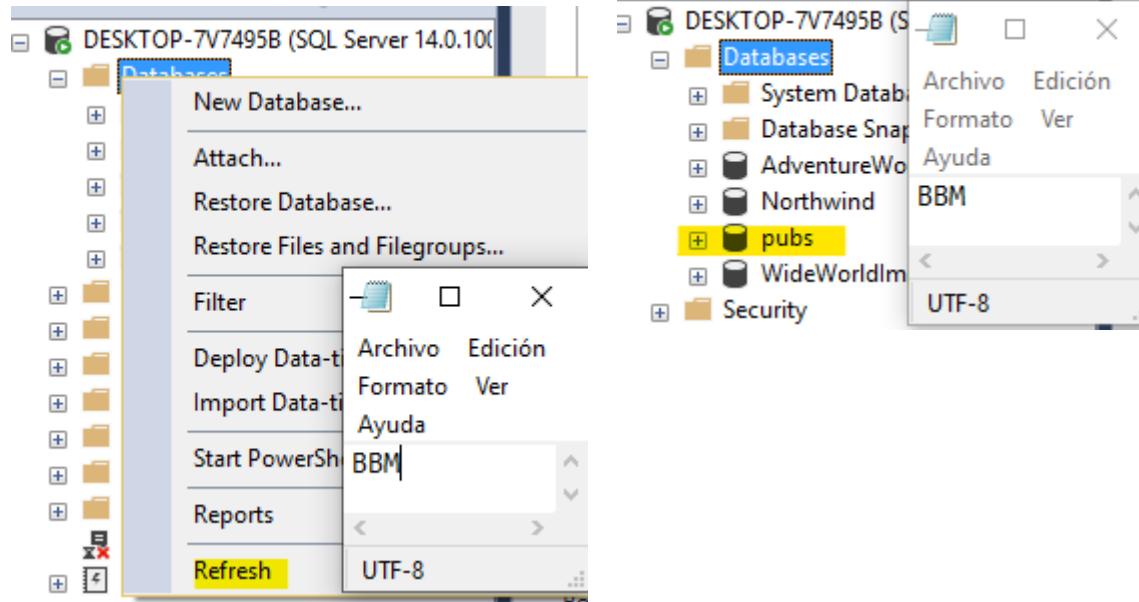
```
--execute sp_dboption 'pubs' , 'trunc. log on chkpt.' , 'true'  
--GO
```

Para acabar seleccionamos todo y le damos al botón **Ejecutar**.



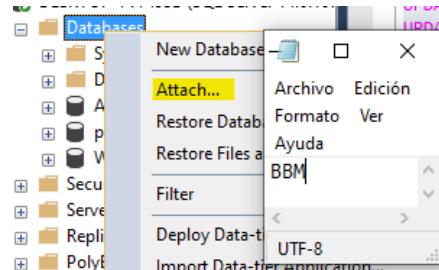
Execute

Cuando acaba, refrescamos y ya vemos la base de datos

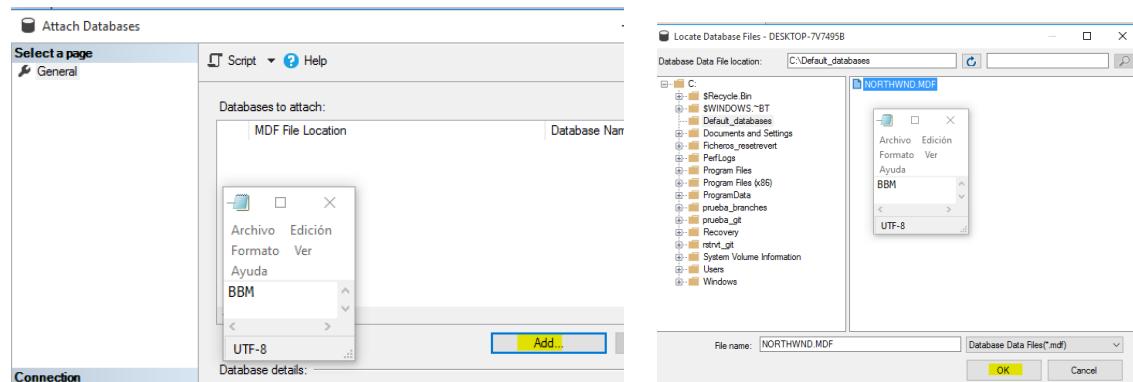


Northwind con ATTACH

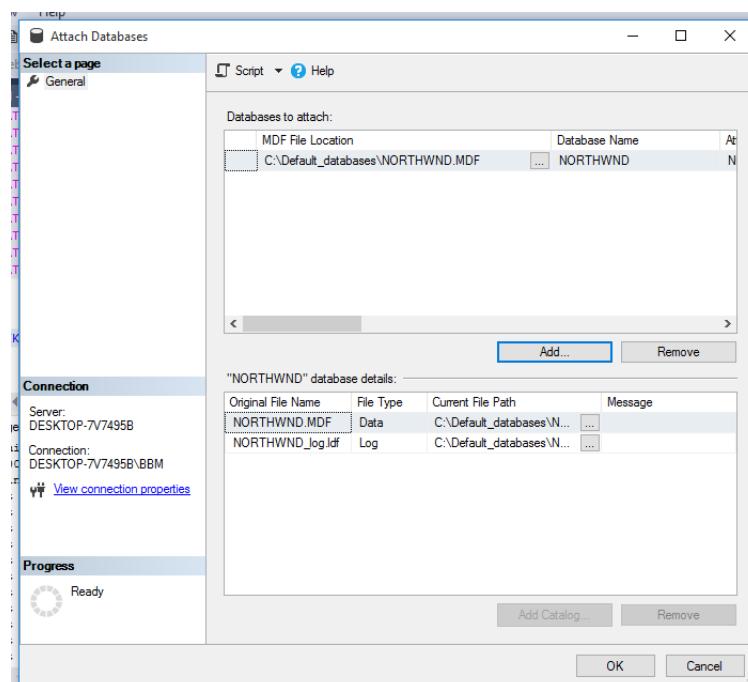
Lo primero que hacemos es clicar con el botón derecho sobre Databases>Attach...



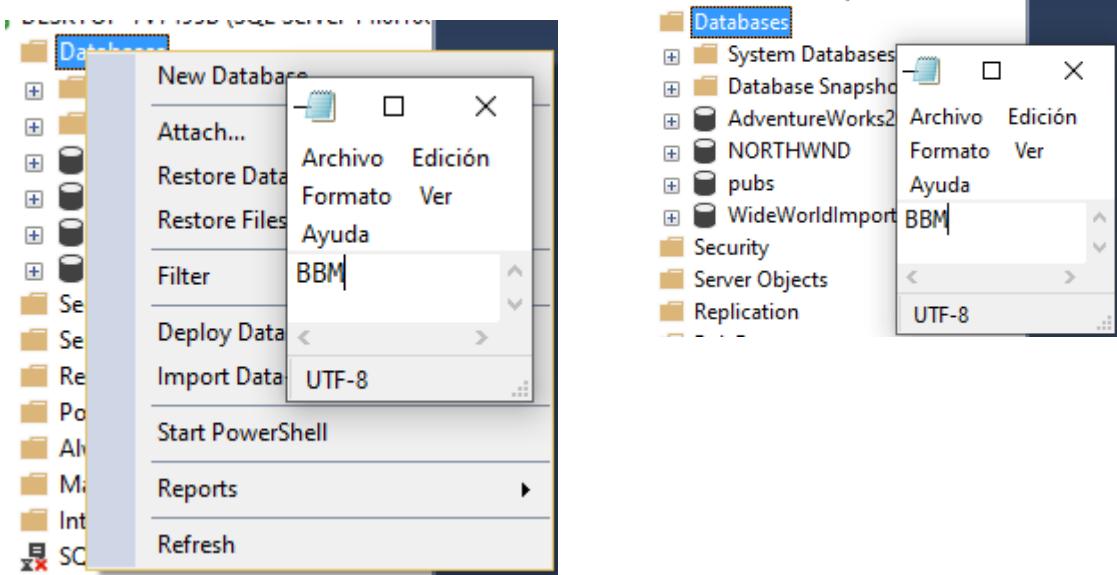
Elegimos la Base de datos que nos interesa en el directorio que la tengamos.



Vemos la base y le damos a Ok (tener en cuenta que tenemos que estar como Administrador)



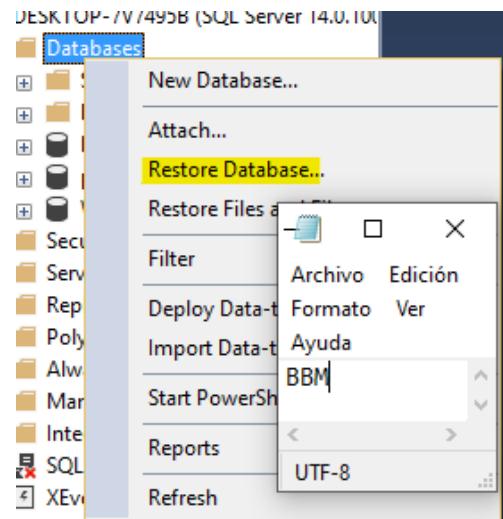
Para acabar refreshamos.



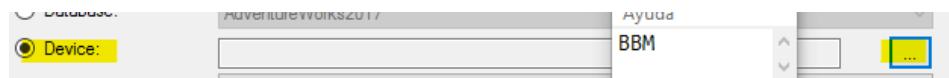
Adventureworks2017 desde BACKUP

Primero descargamos el archivo .bak de la página <https://docs.microsoft.com/es-es/sql/samples/adventureworks-install-configure?view=sql-server-ver15&tabs=ssms#download-backup-files>. En este caso la versión 2017.

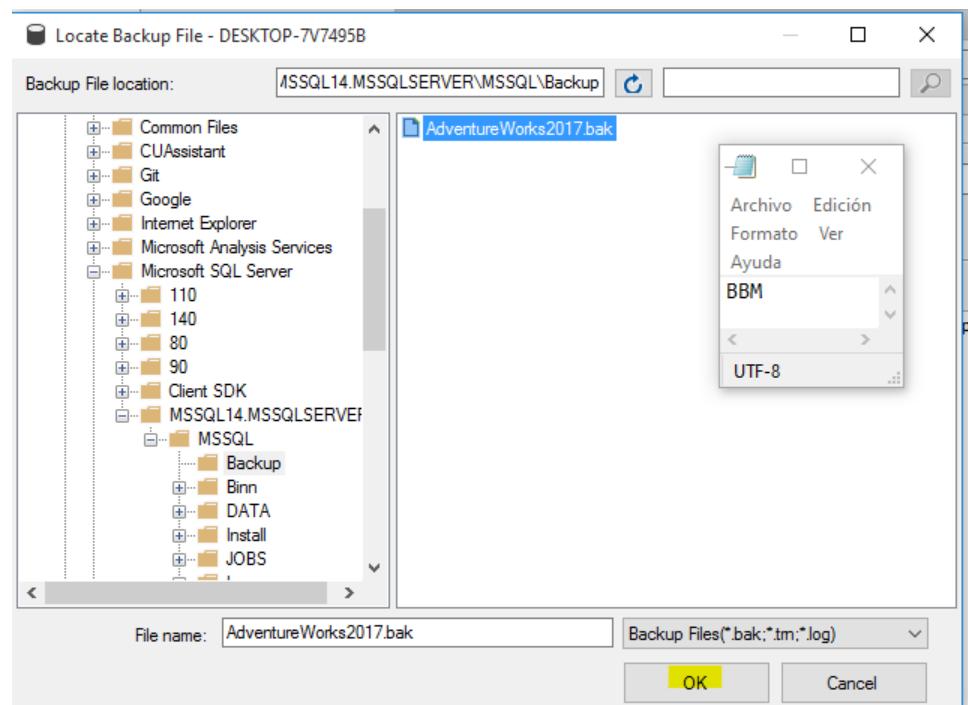
Una vez la tenemos en nuestro sistema, hacemos clic con el botón derecho sobre Databases>Restore Database... para iniciar el asistente.



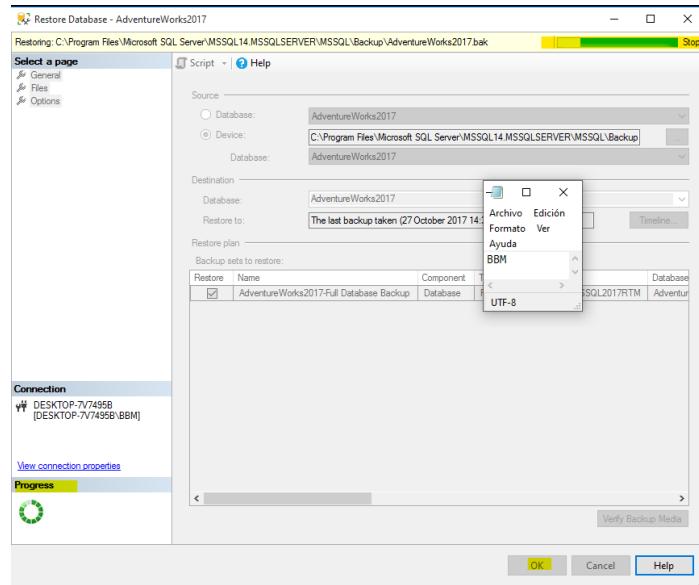
Seleccionamos dispositivo. A continuación, clicamos en los ...



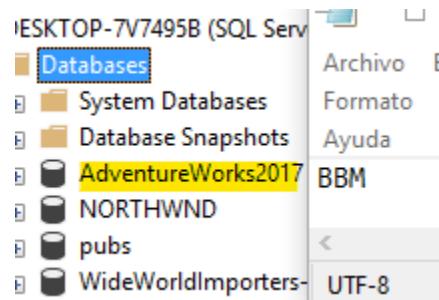
Seleccionamos Add y elegimos el archivo .bak que acabamos de descargar.



Seleccionamos **Aceptar** y comprobamos que tanto la ubicación como los archivos son los correctos para restaurar esta base de datos. Seleccionamos aceptar y la base comenzará a restaurarse. Cuando en el progreso ponga **Sucess**, la base estará restaurada.

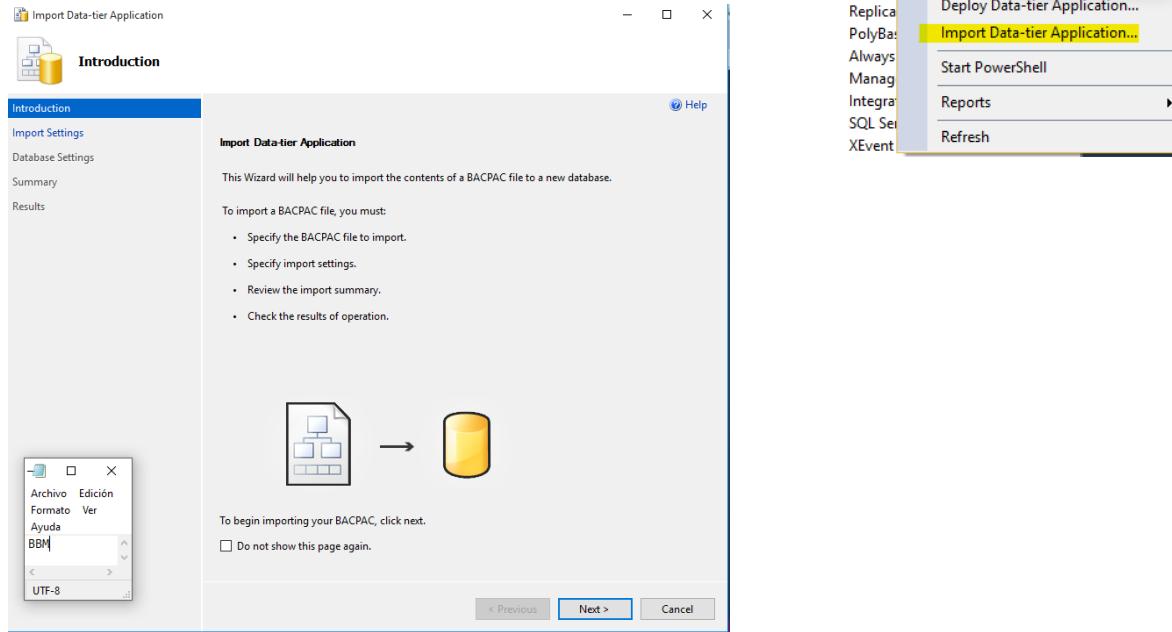


Por último hacemos *Refresh* para ver que la base de datos está disponible.

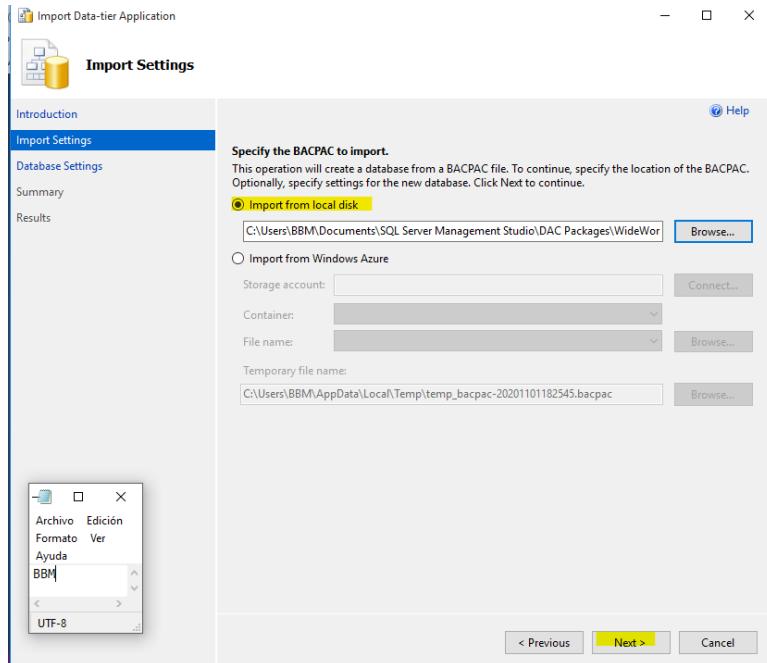


WideWorldImporters con BACPAC

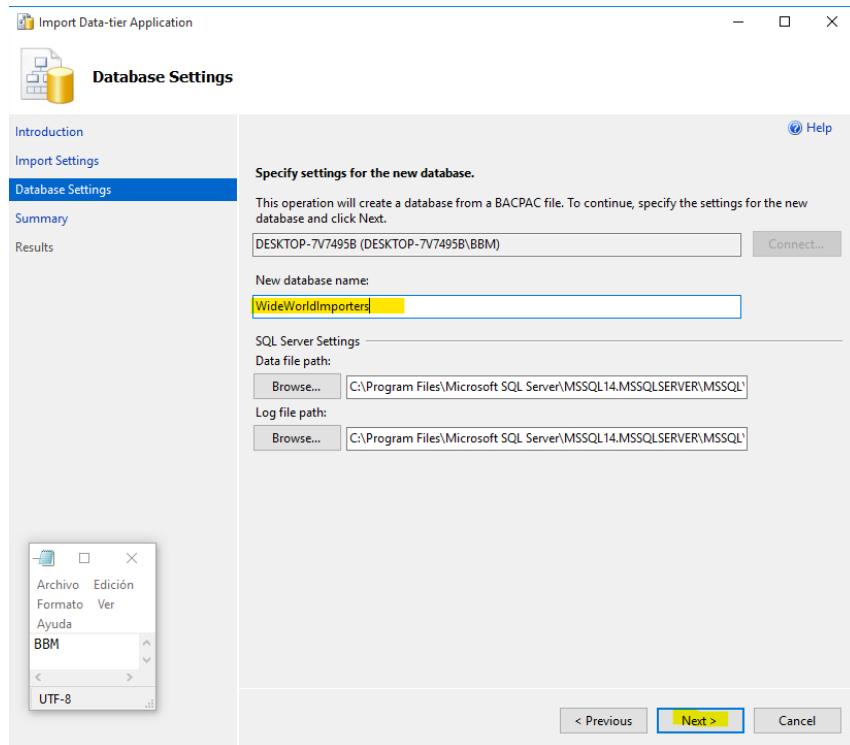
Primero hacemos clic con el botón derecho en **Databases>Import Data-tier applications**. Se nos va a abrir un asistente (*wizard*) cuya primera ventana es de presentación, clicamos *Next*.



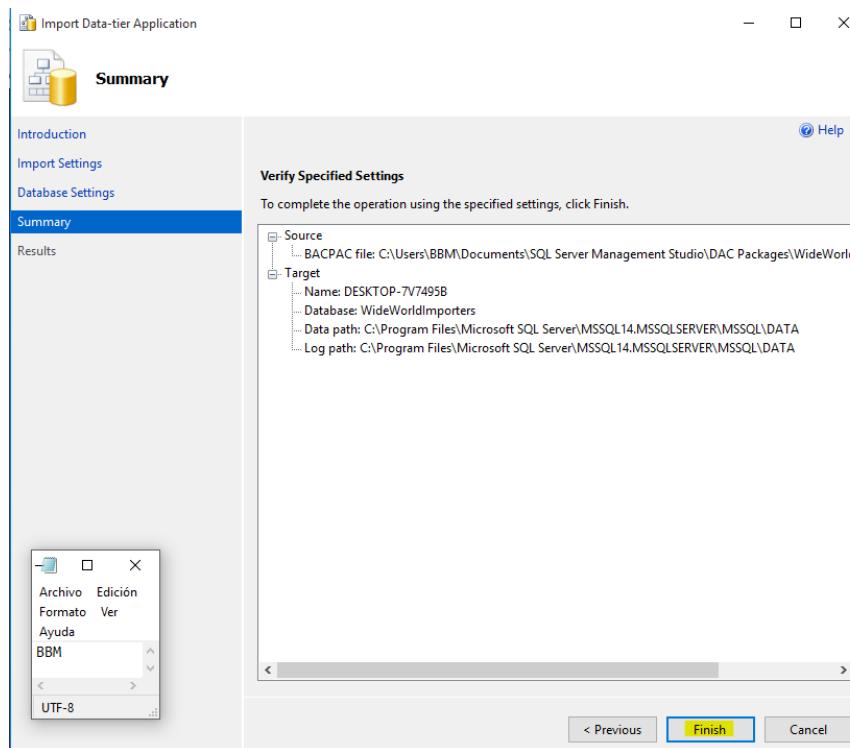
En la siguiente ventana seleccionamos Importar desde un disco local después de buscar el archivo en el directorio que lo tenemos insertado.



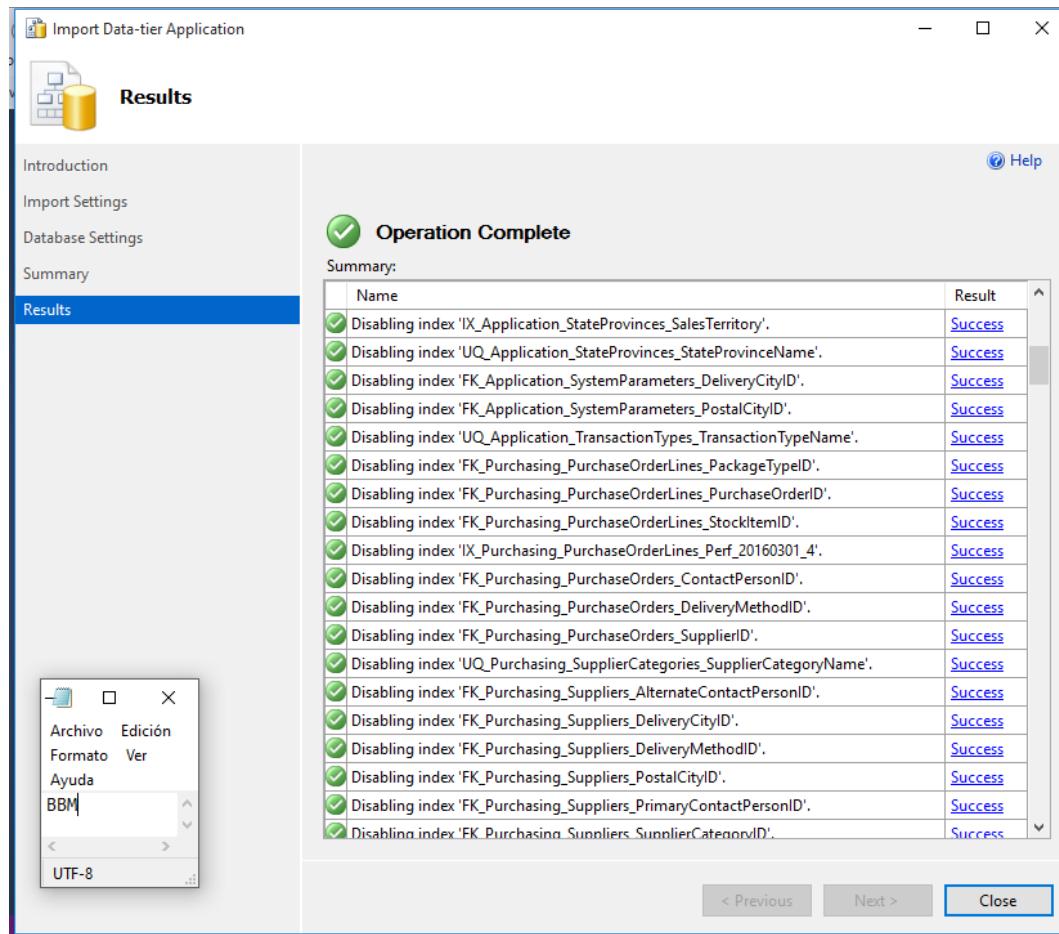
En la siguiente ventana podremos modificarle el nombre. Yo le he puesto *WideWorldImporters* (antes *WideWorldImporters-Standard*).



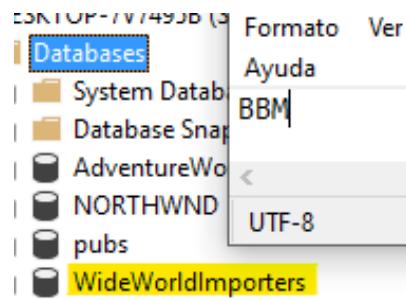
La siguiente ventana es un resumen de lo que hemos elegido.



Una vez acabado nos dirá que la operación se ha completado.



Refrescamos y ya vemos el resultado.



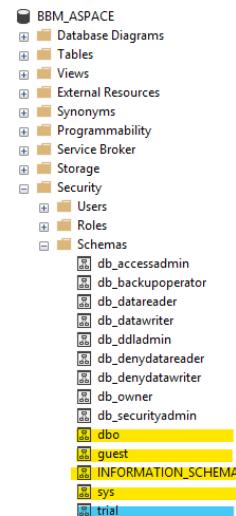
PRUEBAS PREVIAS

Para poder realizar todas las pruebas, a la base de datos creada con el **Data Modeler** le haré lo siguiente. Primero creo un **schema** llamado **trial** dentro de la base de datos para hacer unas pruebas básicas.

Los **esquemas (schema)** en SQL Server se utilizan para **organizar o agrupar los conjuntos de objetos de una base de datos**. Permitiendo una mejor administración de permisos al momento de asignarlos a los usuarios.

Entre los objetos que utilizan esquemas podemos encontrar:

- Tablas
- Diagramas
- Procedimientos almacenados
- Funciones
- Vistas



Una base de datos crea por defecto algunos esquemas que se utilizan al crear objetos, los principales son.

- sys
- guest
- INFORMATION_SCHEMA
- dbo → Es el que entiende por defecto SQL Server

Estos esquemas por defecto (*default*) no se pueden eliminar, y se puede localizar esta información en la carpeta seguridad-Esquemas.

En cuanto a limitaciones y restricciones:

- El esquema nuevo es propiedad de una de las siguientes entidades de seguridad de nivel de base de datos: usuario de base de datos, rol de base de datos o rol de aplicación. Los objetos creados dentro de un esquema son propiedad del esquema y tienen un principal_id NULL en sys.objects. La propiedad de los objetos incluidos en el esquema puede transferirse a cualquier entidad de seguridad de nivel de base de datos, pero el propietario del esquema siempre mantiene el permiso CONTROL en los objetos del esquema.
- Al crear un objeto de base de datos, si especifica una entidad de seguridad de dominio válida (usuario o grupo) como la propietaria del objeto, la entidad de seguridad de dominio se agrega a la base de datos como esquema. Esta entidad de seguridad de dominio es la propietaria del nuevo esquema.

Para la creación del *schema* se utiliza el comando **CREATE SCHEMA <nombre schema>**:

```
USE BBM_ASPACE
GO
/*CREO ESQUEMA PARA HACER PRUEBAS. PRIMERO COMPRUEBO SU EXISTENCIA*/
DROP SCHEMA IF EXISTS trial
GO
```

```
CREATE SCHEMA trial
GO
```

A este *schema* le creo unas **tablas**. Especial atención en que se le especifica el *schema* **[trial] . [nombre_tabla]**:

```
DROP TABLE IF EXISTS trial.BBM_Expediente
GO

CREATE TABLE [trial].[BBM_Expediente](
    [ID_Expediente] [varchar](10) NOT NULL,
    [BBM_Enfermedad_Diagnosticada_Nombre_Enfermedad] [varchar](50) NOT
NULL,
    [BBM_Usuario_Usuario_ID] [varchar](10) NOT NULL,
    [BBM_Sesion_Sesion_ID] [char](14) NOT NULL,
    [BBM_Tratamientos_BBM_Tratamientos_ID] [char](8) NOT NULL,
    CONSTRAINT [BBM_Expediente_PK] PRIMARY KEY CLUSTERED
(
    [ID_Expediente] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =
OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
```

```
-----
DROP TABLE IF EXISTS trial.BBM_Sesion
GO
```

```
CREATE TABLE [trial].[BBM_Session](
    [Sesion_ID] [char](14) NOT NULL,
    [Fecha_y_Hora_de_inicio] [datetime] NULL,
    [Fecha_y_Hora_de_fin] [datetime] NULL,
    [BBM_Horario_ID_Horario] [char](8) NOT NULL,
    [BBM_Espacio_ID_Espacio] [char](10) NOT NULL,
    [BBM_Equipamiento_ID_Equipamiento] [char](10) NOT NULL,
    [BBM_Emppleado_BBM_Emppleado_ID] [int] NOT NULL,
    [BBM_Actividades_BBM_Actividades_ID] [int] NOT NULL,
    CONSTRAINT [BBM_Sesion_PK] PRIMARY KEY CLUSTERED
(
    [Session_ID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =
OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
```

```
-----
DROP TABLE IF EXISTS trial.BBM_Medicacion
GO
```

```
CREATE TABLE [trial].[BBM_Medicacion] (
    [ID_Medicacion] [char](12) NOT NULL,
    [Nombre_Medicamento] [varchar](25) NOT NULL,
    [Codigo_EAN] [int] NOT NULL,
    [Fabricante] [varchar](50) NULL,
    [Posología] [varchar](255) NULL,
    [Descripción] [varchar](255) NULL,
    [Observaciones] [varchar](255) NULL,
    CONSTRAINT [BBM_Medicacion_PK] PRIMARY KEY CLUSTERED
(
    [ID_Medicacion] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =
OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
DROP TABLE IF EXISTS trial.BBM_Comidas
GO

CREATE TABLE [trial].[BBM_Comidas] (
    [ID_Menu] [char](9) NOT NULL,
    [Plato_1] [varchar](255) NULL,
    [Plato_2] [varchar](255) NULL,
    [Plato_3] [varchar](255) NULL,
    [BBM_Otros_Departamentos_OD_ID] [char](5) NOT NULL,
    CONSTRAINT [BBM_Comidas_PK] PRIMARY KEY CLUSTERED
(
    [ID_Menu] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =
OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
-----
----  

DROP TABLE IF EXISTS trial.BBM_Enfermedad_Diagnosticada
GO

CREATE TABLE [trial].[BBM_Enfermedad_Diagnosticada] (
    [Nombre_Enfermedad] [varchar](50) NOT NULL,
    [Nombre_Comun_Enfermedad] [varchar](50) NOT NULL,
    [Descripción] [varchar](255) NULL,
    [Síntomas] [varchar](255) NULL,
    CONSTRAINT [BBM_Enfermedad_Diagnosticada_PK] PRIMARY KEY CLUSTERED
(
    [Nombre_Enfermedad] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =
OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
-----
----  

DROP TABLE IF EXISTS trial.BBM_Tratamientos
GO

CREATE TABLE [trial].[BBM_Tratamientos] (
    [BBM_Tratamientos_ID] [char](8) NOT NULL,
    [Tratamiento_ID] [char](4) NOT NULL,
    [BBM_Auxiliar_BBM_Emppleado_BBM_Emppleado_ID] [int] NULL,
```

```

    [BBM_Especialista_BBMM_Emppleado_BBMM_Emppleado_ID] [int] NULL,
  CONSTRAINT [BBM_Tratamientos_PK] PRIMARY KEY CLUSTERED
  (
    [BBM_Tratamientos_ID] ASC
  )WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =
OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
-----
DROP TABLE IF EXISTS trial.BBM_Usuario
GO

CREATE TABLE [trial].[BBM_Usuario](
  [Usuario_ID] [varchar](10) NOT NULL,
  [Nombre_1] [varchar](20) NULL,
  [Nombre_2] [varchar](20) NULL,
  [Apellido_1] [varchar](20) NULL,
  [Apellido_2] [varchar](20) NULL,
  [DNI] [varchar](9) NOT NULL,
  [Otros_Detalles] [varchar](255) NULL,
  CONSTRAINT [BBM_Usuario_PK] PRIMARY KEY CLUSTERED
  (
    [Usuario_ID] ASC
  )WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =
OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
-----
DROP TABLE IF EXISTS trial.BBM_Especialista
GO

CREATE TABLE trial.BBM_Especialista
(
  Especialista_ID VARCHAR (5) NOT NULL ,
)
GO

ALTER TABLE trial.BBM_Especialista ADD CONSTRAINT Especialista_ID_PK
PRIMARY KEY CLUSTERED (Especialista_ID)
  WITH (
  ALLOW_PAGE_LOCKS = ON ,
  ALLOW_ROW_LOCKS = ON )
GO
-----
```

A estas tablas dentro del *schema trial* le inserto unos valores. Para ello se utiliza la palabra reservada **INSERT INTO [schema] . [tabla]**

(campo1,campo2,etc)

Palabra reservada VALUES

```
("valor_campo_1","valor_campo_2","etc")

--TRATAMIENTOS
INSERT INTO [trial].[BBM_Tratamientos]

([BBM_Tratamientos_ID],[Tratamiento_ID],[BBM_Auxiliar_BBM_Empleado_BBM_Employeado_ID],[BBM_Especialista_BBM_Empleado_BBM_Empleado_ID])
VALUES
('FISI','FISI','123','','')
GO
INSERT INTO [trial].[BBM_Tratamientos]

([BBM_Tratamientos_ID],[Tratamiento_ID],[BBM_Auxiliar_BBM_Empleado_BBM_Employeado_ID],[BBM_Especialista_BBM_Empleado_BBM_Empleado_ID])
VALUES
('TERO','141','','')
GO
INSERT INTO [trial].[BBM_Tratamientos]

([BBM_Tratamientos_ID],[Tratamiento_ID],[BBM_Auxiliar_BBM_Empleado_BBM_Employeado_ID],[BBM_Especialista_BBM_Empleado_BBM_Empleado_ID])
VALUES
('PSIC','235','','')
GO
-----
--USUARIO
INSERT INTO [trial].[BBM_Usuario]

([Usuario_ID],[Nombre_1],[Nombre_2],[Apellido_1],[Apellido_2],[DNI],[Otros_Detalles])
VALUES
('USERGPP012','PACO','','GONZALEZ','PEREZ','11111012S','')
GO
INSERT INTO [trial].[BBM_Usuario]

([Usuario_ID],[Nombre_1],[Nombre_2],[Apellido_1],[Apellido_2],[DNI],[Otros_Detalles])
VALUES
('USERVNA013','ANA','MARIA','VAZQUEZ','NUNEZ','11111013N','')
GO
INSERT INTO [trial].[BBM_Usuario]

([Usuario_ID],[Nombre_1],[Nombre_2],[Apellido_1],[Apellido_2],[DNI],[Otros_Detalles])
VALUES
('USERVRI013','ISABEL','','VILLAR','RODRIGUEZ','11111013D','Violento')
GO
-----
--ENFERMEDAD DIAGNOSTICADA
INSERT INTO [trial].[BBM_Enfermedad_Diagnosticada]
```

```
([Nombre_Enfermedad],[Nombre_Comun_Enfermedad],[Descripcion],[Sintomas])
      VALUES
```

('APRAXIA IDEOMOTORA','APRAXIA','Incapacidad para realizar tareas que requieren recordar patrones o secuencias de movimientos.', 'no pueden recordar ni hacer la secuencia de movimientos necesaria para completar habilidades sencillas o tareas complejas, a pesar de que tienen la capacidad fca para realizar la tarea y de que pueden hacer los movimientos simples de la tarea.')

```
GO
```

```
INSERT INTO [trial].[BBM_Enfermedad_Diagnosticada]
```

```
([Nombre_Enfermedad],[Nombre_Comun_Enfermedad],[Descripcion],[Sintomas])
      VALUES
```

('PARALISIS CEREBRAL','PA','Lesión en el cerebro que afecta a la movilidad y la postura de la persona, limitando su actividad.', 'VARIOS')

```
GO
```

--COMIDAS

```
INSERT INTO [trial].[BBM_Comidas]
```

```
([ID_Menu],[Plato_1],[Plato_2],[Plato_3],[BBM_Otros_Departamentos_OD_ID])
      VALUES
```

('DESAYUNO','PORRIDGE DE AVENA','','','')

```
GO
```

```
INSERT INTO [trial].[BBM_Comidas]
```

```
([ID_Menu],[Plato_1],[Plato_2],[Plato_3],[BBM_Otros_Departamentos_OD_ID])
      VALUES
```

('COMIDA','SOPA','FILETE DE TERNERA A LA PLANCHA CON BROCOLI AL VAPOR','MANZANA','')

```
GO
```

--MEDICACION

```
INSERT INTO [trial].[BBM_Medicacion]
```

```
([ID_Medicacion],[Nombre_Medicamento],[Codigo_EAN],[Fabricante],[Posologí a],[Descripcion],[Observaciones])
```

```
      VALUES
```

('NEUNUFOR','NEURO-NUTRITION FORMULA',' ','superSMART','Tomar de 1 a 3 cucharadas vegetales al dcon las comidas.', '','Contiene derivados de la soja')

```
GO
```

-- SESION

```
INSERT INTO [trial].[BBM_Sesion]
```

```
([Sesion_ID],[Fecha_y_Hora_de_inicio],[Fecha_y_Hora_de_fin],[BBM_Horario_ID_Horario],[BBM_Espacio_ID_Espacio],[BBM_Equipamiento_ID_Equipamiento],[BBM_Empleado_BBM_Empleado_ID],[BBM_Actividades_BBM_Actividades_ID])
```

```
      VALUES
```

```

('LU25012021','01/25/2021 10:00','01/25/2021
11:00','','','INDIBA','','')
GO
INSERT INTO [trial].[BBM_Sesion]
([Sesion_ID],[Fecha_y_Hora_de_inicio],[Fecha_y_Hora_de_fin],[BBM_Horario_
ID_Horario],[BBM_Espacio_ID_Espacio],[BBM_Equipamiento_ID_Equipamiento],[
BBM_Emppleado_BBM_Emppleado_ID],[BBM_Actividades_BBM_Actividades_ID])
VALUES
('DO24012021','01/24/2021 10:00','01/24/2021
12:00','','','ELECTRO','','')
GO
-----
--EXPEDIENTE
INSERT INTO [trial].[BBM_Expediente]
([ID_Expediente],[BBM_Enfermedad_Diagnosticada_Nombre_Enfermedad],[BBM_Us
uario_Usuario_ID],[BBM_Sesion_Sesion_ID],[BBM_Tratamientos_BBM_Tratamient
os_ID])
VALUES
('GPPTUT012','APRAXIA
IDEOMOTORA','USERGPP012','LU25012021','PSIC')
GO
-----
--ESPECIALISTA
INSERT INTO [trial].[BBM_Especialista]
([Especialista_ID])
VALUES
('FISIO')
GO

```

Como es una prueba real, le añado a mano las relaciones que tiene la base de datos. Dichas relaciones son las *primary key* y *foreign key* de las tablas dbo que tengo creadas, pero para hacer la prueba se las añado aquí.

Las **claves primarias** o **PRIMARY KEY**, identifican de manera única cada fila de una tabla. La columna definida como clave primaria (**PRIMARY KEY**) debe ser **UNIQUE** (valor único) y **NOT NULL** (no puede contener valores nulos).

NOTA: Cada tabla sólo puede tener una clave primaria (**PRIMARY KEY**). También puede estar compuesta por varias columnas, por ejemplo por las columnas 'identificador' y 'nombre'

La **clave externa** o **FOREIGN KEY**, es una columna o varias columnas, que sirven para señalar cual es la clave primaria de otra tabla. La columna o columnas señaladas como **FOREIGN KEY**, solo podrán tener valores que ya existan en la clave primaria **PRIMARY KEY** de la otra tabla.

```
-- PARA UNA PRUEBA MÁS REAL, AÑADIMOS RELACIONES CON FK

ALTER TABLE [trial].[BBM_Expediente]
    ADD CONSTRAINT BBM_Enfermedad_Diagnosticada_Nombre_Enfermedad_FK
    FOREIGN KEY (BBM_Enfermedad_Diagnosticada_Nombre_Enfermedad)
        REFERENCES trial.BBM_Enfermedad_Diagnosticada (Nombre_Enfermedad)
        ON DELETE CASCADE
        ON UPDATE CASCADE
;
ALTER TABLE [trial].[BBM_Expediente]
    ADD CONSTRAINT BBM_Usuario_Usuario_ID_FK FOREIGN KEY
(BBM_Usuario_Usuario_ID)
    REFERENCES trial.BBM_Usuario (Usuario_ID)
    ON DELETE CASCADE
    ON UPDATE CASCADE
;
ALTER TABLE [trial].[BBM_Expediente]
    ADD CONSTRAINT BBM_Sesion_Sesion_ID_FK FOREIGN KEY
(BBM_Sesion_Sesion_ID)
    REFERENCES trial.BBM_Sesion (Sesion_ID)
    ON DELETE CASCADE
    ON UPDATE CASCADE
;
ALTER TABLE [trial].[BBM_Expediente]
    ADD CONSTRAINT BBM_Tratamientos_BBM_Tratamientos_ID_FK FOREIGN KEY
(BBM_Tratamientos_BBM_Tratamientos_ID)
    REFERENCES trial.BBM_Tratamientos (BBM_Tratamientos_ID)
    ON DELETE CASCADE
    ON UPDATE CASCADE
;
ALTER TABLE [trial].[BBM_Expediente]
    ADD Especialista_ID VARCHAR (5) NULL;
GO

ALTER TABLE [trial].[BBM_Expediente]
    ADD CONSTRAINT Especialista_ID_FK FOREIGN KEY (Especialista_ID)
    REFERENCES trial.BBM_Especialista (Especialista_ID)
    ON DELETE CASCADE
    ON UPDATE CASCADE
;
-----
```

Una vez insertados todos los valores, pruebo a crear una *vista* para el usuario. En ella, solo quiero que los usuarios vean su número de expediente, la enfermedad diagnosticada y su propio ID para que sepan que están consultando el expediente correcto.

Una *vista* es una tabla virtual cuyo contenido está definido por una consulta. Al igual que una tabla, una *vista* consta de un conjunto de columnas y filas de datos con un

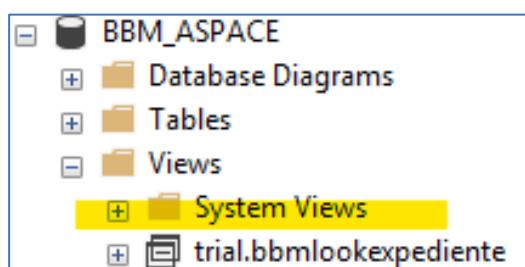
nombre. Sin embargo, a menos que esté indizada, una *vista* no existe como conjunto de valores de datos almacenados en una base de datos. **Las filas y las columnas de datos proceden de tablas a las que se hace referencia en la consulta que define la vista** y se producen de forma dinámica cuando se hace referencia a la vista.

Una vista actúa como filtro de las tablas subyacentes a las que se hace referencia en ella. La consulta que define la vista puede provenir de una o de varias tablas, o bien de otras vistas de la base de datos actual u otras bases de datos. Asimismo, es posible utilizar las consultas distribuidas para definir vistas que utilicen datos de orígenes heterogéneo.

Las vistas suelen usarse para centrar, simplificar y personalizar la percepción de la base de datos para cada usuario. Las vistas pueden emplearse como mecanismos de seguridad, que permiten a los usuarios obtener acceso a los datos por medio de la vista, pero no les conceden el permiso de obtener acceso directo a las tablas base subyacentes de la vista.

Además del rol estándar de las vistas básicas definidas por el usuario, SQL Server proporciona los siguientes tipos de vistas que permiten llevar a cabo objetivos especiales en una base de datos.

- **Vistas indizadas** → vista que se ha **materializado**. Esto significa que se ha calculado la definición de la vista y que los datos resultantes se han almacenado como una tabla. Se puede indizar una vista creando un índice clúster único en ella. Las vistas indizadas pueden mejorar de forma considerable el rendimiento de algunos tipos de consultas. Las vistas indizadas funcionan mejor para consultas que agregan muchas filas. No son adecuadas para conjuntos de datos subyacentes que se actualizan frecuentemente.
- **Vistas con particiones** → combina datos horizontales con particiones de un conjunto de tablas miembro en uno o más servidores. Esto hace que los datos aparezcan como si fueran de una tabla. Una vista que combina tablas **miembro** en la misma instancia de SQL Server es una vista con particiones local.
- **Vistas del sistema** → exponen metadatos de catálogo. Puede usar las vistas del sistema para devolver información acerca de la instancia de SQL Server u objetos definidos en la instancia. Por ejemplo, puede consultar la vista de catálogo **sys.databases** para devolver información sobre las bases de datos definidas por el usuario disponibles en la instancia.



```

/*CREAMOS UNA VISTA PARA EL USUARIO*/
SELECT *
FROM trial.BBM_Expediente
GO
--ID_Expediente BBM_Enfermedad_Diagnosticada_Nombre_Enfermedad
BBM_Usuario_Usuario_ID
--GPPTUT012      APRAXIA IDEOMOTORA
USERGPP012
/*QUIERO QUE SOLO VEA LOS CAMPOS DE ARRIBA*/

/*CREO LA VISTA CONTROLANDO PRIMERO SU EXISTENCIA*/
DROP VIEW IF EXISTS trial.bbmlookexpediente
GO

CREATE VIEW trial.bbmlookexpediente
AS
SELECT
    ID_Expediente, BBM_Enfermedad_Diagnosticada_Nombre_Enfermedad,
    BBM_Usuario_Usuario_ID
FROM trial.BBM_Expediente
GO

/*CREO UN ROL*/
DROP ROLE IF EXISTS BBMConsultasUsuarios
GO

CREATE ROLE BBMConsultasUsuarios
GO

```

Ahora para que los usuarios tengan permisos, lo que hago es: primero concedo permisos al rol:

```

/*DOY PERMISOS DE SELECT AL ROL*/
GRANT SELECT ON trial.bbmlookexpediente TO BBMConsultasUsuarios;
GO

```

Creo un usuario. Para crear un usuario es **CREATE USER <usuario> WITHOUT LOGIN**

```

/*CREAMOS UN USUARIO PARA ASIGNARLE EL ROL*/

DROP USER IF EXISTS USERGPP012
GO
CREATE USER USERGPP012 WITHOUT LOGIN;
GO

```

Asigno el usuario al rol:

```

-- METEMOS AL USUARIO EN EL ROL
ALTER ROLE BBMConsultasUsuarios
ADD MEMBER USERGPP012;
GO

```

Ahora hago pruebas con el usuario: Me impersono con el comando **EXECUTE AS USER = '<usuario>'**. Luego compruebo con **PRINT USER** quién soy y ejecuto una consulta:

```
/*NOS IMPERSONAMOS COMO EL USUARIO USERGPP012 PARA QUE SE VEA QUE PUEDE  
HACER CONSULTAS SOBRE LA VISTA PERMITIDA EN EL ROL*/  
  
EXECUTE AS USER = 'USERGPP012';  
GO  
  
PRINT USER  
GO  
  
SELECT * FROM trial.bbbmlookexpediente;  
GO  
  
REVERT  
  
PRINT USER  
GO
```

Ahora ejecuto la misma prueba, pero con procedimiento almacenado. Un **procedimiento almacenado de SQL Server** es un grupo de una o más instrucciones Transact-SQL y se asemejan a las construcciones de otros lenguajes de programación, porque pueden:

- Aceptar parámetros de entrada y devolver varios valores en forma de parámetros de salida al programa que realiza la llamada.
- Contener instrucciones de programación que realicen operaciones en la base de datos. Entre otras, pueden contener llamadas a otros procedimientos.
- Devolver un valor de estado a un programa que realiza una llamada para indicar si la operación se ha realizado correctamente o se han producido errores, y el motivo de estos.

Existen varios tipos de procedimiento almacenado:

- **Definidos por el usuario** → Un procedimiento definido por el usuario se puede crear en una base de datos definida por el usuario o en todas las bases de datos del sistema excepto en la base de datos **Resource**.
- **Temporales** → Los procedimientos temporales son una forma de procedimientos definidos por el usuario. Los procedimientos temporales son iguales que los procedimientos permanentes salvo porque se almacenan en

tempdb. Hay dos tipos de procedimientos temporales (se diferencian entre sí por los nombres, la visibilidad y la disponibilidad):

- Los procedimientos **temporales locales** tienen como primer carácter de sus nombres un solo signo de número (#); solo son visibles en la conexión actual del usuario y se eliminan cuando se cierra la conexión.
- Los procedimientos **temporales globales** presentan dos signos de número (##) antes del nombre; son visibles para cualquier usuario después de su creación y se eliminan al final de la última sesión en la que se usa el procedimiento.
- **Sistema** → Los procedimientos del sistema **se incluyen con SQL Server**. Están almacenados físicamente en la base de datos interna y oculta **Resource** y se muestran de forma lógica en el esquema **sys** de cada base de datos definida por el sistema y por el usuario. Además, la base de datos **msdb** también contiene procedimientos almacenados del sistema en el esquema **dbo** que se usan para programar alertas y trabajos. Dado que los procedimientos del sistema empiezan con el prefijo **sp_**, Windows recomienda que no use este prefijo cuando asigne un nombre a los procedimientos definidos por el usuario.
- SQL Server admite los procedimientos del sistema que proporcionan una interfaz de SQL Server a los programas externos para varias actividades de mantenimiento. Estos procedimientos extendidos usan el prefijo **xp_**.
- **Extendidos definidos por el usuario** → Los procedimientos extendidos le permiten crear sus propias rutinas externas en un lenguaje de programación como puede ser C. Estos procedimientos son **DLL** que una instancia de SQL Server puede cargar y ejecutar dinámicamente.

Los procedimientos almacenados se crean mediante la sentencia CREATE PROCEDURE y se ejecutan con EXEC (o EXECUTE). La sintaxis para crear un procedimiento almacenado con código SQL es la siguiente:

```
create proc NOMBREPROCEDIMIENTO  
@NOMBREPARAMETRO TIPO DATO
```

as

SENTENCIAS SQL;

GO

/*MISMO SCRIPT CON PROCEDIMIENTO ALAMACENADO*/

```

CREATE OR ALTER PROC trial.BBMPprocedure
--PARAMETROS
@ID_Expediente varchar(10) NULL,
@BBM_Enfermedad_Diagnosticada_Nombre_Enfermedad varchar(50) NULL,
@BBM_Usuario_Usuario_ID varchar(10) NULL,
@BBM_Sesion_Sesion_ID char(14) NULL,
@BBM_Tratamientos_BBM_Tratamientos_ID char(8) NULL,
@Especialista_ID varchar(5) NULL
AS
BEGIN
    INSERT INTO trial.BBM_Expediente
        (ID_Expediente,
        BBM_Enfermedad_Diagnosticada_Nombre_Enfermedad, BBM_Usuario_Usuario_ID,
        BBM_Sesion_Sesion_ID, BBM_Tratamientos_BBM_Tratamientos_ID,
        Especialista_ID)
    VALUES
        (@ID_Expediente,
        @BBM_Enfermedad_Diagnosticada_Nombre_Enfermedad, @BBM_Usuario_Usuario_ID,
        @BBM_Sesion_Sesion_ID, @BBM_Tratamientos_BBM_Tratamientos_ID,
        @Especialista_ID);
END;
GO

```

The screenshot shows a SQL query window with the following content:

```

SELECT * FROM sys.objects
WHERE type LIKE 'P'
GO

```

The results grid displays the following data:

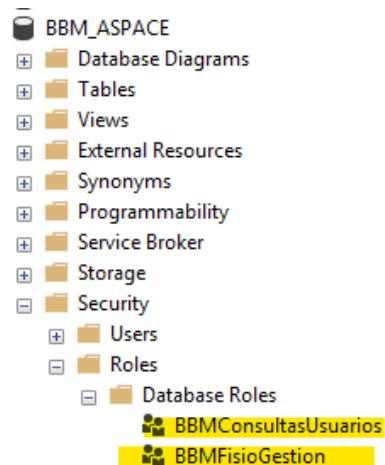
name	object_id	principal_id	schema_id	parent_object_id	type	type_desc	create_date	modify_date	is_ms_shipped	is_published	is_schema_published
BBM_BACKUP_PROCEDURE	935674381	NULL	1	0	P	SQL_STORED_PROCEDURE	2021-01-30 22:11:46.340	2021-01-30 22:11:46.340	0	0	0
BBM_CURSORBACKUP	1575676661	NULL	1	0	P	SQL_STORED_PROCEDURE	2021-02-28 22:06:44.033	2021-02-28 22:06:44.033	0	0	0
BACKUP_ALL_DB_PARENTRADA	1607676775	NULL	1	0	P	SQL_STORED_PROCEDURE	2021-02-28 22:18:17.783	2021-02-28 22:18:17.783	0	0	0
BBM_SPBACKUP	1703677117	NULL	1	0	P	SQL_STORED_PROCEDURE	2021-02-28 22:24:07.850	2021-02-28 22:24:07.850	0	0	0
sp_MSrepl_startup	2123154609	NULL	1	0	P	SQL_STORED_PROCEDURE	2017-08-22 19:40:17.870	2017-08-22 19:40:17.870	1	0	0
sp_MScleanupmergepublisher	2139154666	NULL	1	0	P	SQL_STORED_PROCEDURE	2017-08-22 19:40:17.880	2017-08-22 19:40:17.880	1	0	0

El sistema compila información y la almacena. Siempre queda a la espera. Luego lo llamo a ejecución donde quiero. Para ello, lo voy a demostrar primero creando el rol. Para administrar con facilidad los permisos en las bases de datos, SQL Server proporciona varios *roles*, que son las entidades de seguridad que agrupan a otras entidades de seguridad. Son como los **grupos** del sistema operativo Microsoft Windows. Los roles de nivel de base de datos se aplican a toda la base de datos en lo que respecta a su ámbito de permisos.

Existen dos tipos de roles en el nivel de base de datos: los **roles fijos de base de datos** que están predefinidos en la base de datos y los **roles de base de datos definidos por el usuario** que el usuario puede crear.

Los roles fijos de base de datos se definen en el nivel de base de datos y existen en cada una de ellas. Los miembros de los roles de base de datos `db_owner` pueden administrar la pertenencia a roles fijos de base de datos. También hay algunos roles de base de datos con fines especiales en la base de datos `msdb`.

```
lo llamas a ejecución desde donde quieras
/*****DEMOSTRACIÓN*****
DROP ROLE IF EXISTS BBMFisiogestion
GO
CREATE ROLE BBMFisiogestion
GO
```



Concedo **permisos sobre el rol**. Los permisos de los roles de base de datos definidos por el usuario se pueden personalizar con las instrucciones `GRANT`, `DENY` y `REVOKE`

```
--El sistema compila info y la almacena. Siempre queda a la espera. Luego
t
GRANT EXECUTE ON SCHEMA::[trial] TO BBMFisiogestion; --> DOY LOS PERMISOS
DE SCHEMA AL ROL
GO
```

Creo un usuario. Para agregar y quitar usuarios en un rol de base de datos, use las opciones `ADD MEMBER` y `DROP MEMBER` de la instrucción `ALTER ROLE` .

```
CREATE USER FISIOESPECIALISTA WITHOUT LOGIN;
GO
```

Añado el usuario al rol:

```
ALTER ROLE BBMFisiogestion --> AGDO EL MIEMBRO AL ROL
ADD MEMBER FISIOESPECIALISTA
GO
```

Ahora, con el procedimiento almacenado creado anterior, hago que el usuario creado intente hacer una inserción, pero en vez de hacerlo manualmente, lo hace a través de una **ejecución**. Para ello, me impersono en él y lo pruebo:

--> HACEMOS QUE EL FISIOESPECIALISTA INTENTE HACER UNA INSERCIM. NO DE MANERA MANUAL, PUES NO TIENE PERMISOS, SINO A TRAVÉS DEL PROCEDIMIENTO ALMACENADO QUE ACABAMOS DE CREAR

```
EXECUTE AS USER = 'FISIOESPECIALISTA'
GO

PRINT USER
GO

EXEC trial.BBMProcedure
--PARAMETROS
@ID_Expediente = 'TESTEO001',
@BBM_Enfermedad_Diagnosticada_Nombre_Enfermedad = 'APRAXIA
IDEOMOTORA',
@BBM_Usuario_Usuario_ID = 'USERGPP012',
@BBM_Sesion_Sesion_ID = 'DO24012021',
@BBM_Tratamientos_BBM_Tratamientos_ID = 'FISI      ',
@Especialista_ID = 'FISIO';
GO

REVERT

PRINT USER
GO

SELECT *
FROM trial.BBM_Expediente
GO
```

FILEGROUP

Para las pruebas que vienen a continuación, como son pruebas que no quiero que afecten a mi base de datos antes de hacer una implementación, **aislé los schemas trial en una base de datos aparte**. Para ello, primero creé el **Filegroup** de la siguiente manera.

Como mínimo, todas las bases de datos de SQL Server tienen dos archivos del sistema operativo: un **archivo de datos** y un **archivo de registro**. Los archivos de datos contienen datos y otros objetos, como tablas, índices, procedimientos almacenados y vistas.

- Los archivos de registro contienen la información necesaria para recuperar todas las transacciones de la base de datos.
- Los archivos de datos se pueden agrupar en grupos de archivos para su asignación y administración.

SQL Server tienen tres tipos de archivos:

- **Principal** → Incluye información de inicio de la base de datos y apunta a los demás archivos de la misma. Cada base de datos tiene un archivo de datos principal. La extensión recomendada para los nombres de archivos de datos principales es **.mdf**.
- **Secundario** → Archivos de datos opcionales definidos por el usuario. Los datos se pueden distribuir en varios discos colocando cada archivo en una unidad de disco distinta. La extensión de nombre de archivo recomendada para los archivos de datos secundarios es **.ndf**.
- **Registro de transacciones** → El registro contiene la información que se utiliza para recuperar la base de datos. Cada base de datos debe tener al menos un archivo de registro. La extensión recomendada para los nombres de archivos de registro es **.ldf**.

Primero creo la base de datos con tan sólo los archivos **.mdf** y **.ldf**:

Ahora **alter** en la base y voy añadiendo las **organizaciones lógicas de archivos físicos (.ndf)**:

Primero controlo la existencia de la base de datos.

```
USE master  
GO
```

```
DROP DATABASE IF EXISTS TRIALBBM_INCLUDENDF_INCLUDENDF  
GO
```

Posteriormente creo la base de datos

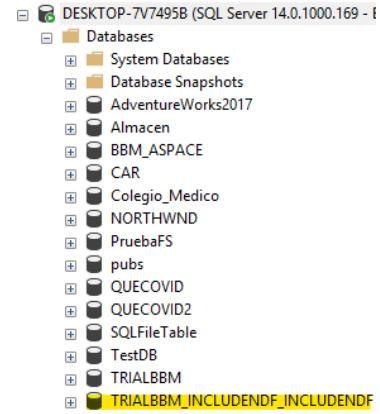
```

CREATE DATABASE TRIALBBM_INCLUDENDF_INCLUDENDF
    ON
        PRIMARY
            (
                NAME = BBM_ASPACE ,
                FILENAME =
                    'C:\BBMTRIGGERS_TRIAL\BBM_ASPACE_MDF.mdf' ,
                SIZE = 10MB ,
                MAXSIZE = 50 ,
                FILEGROWTH = 5% ) ,
            (
                NAME = BBM_Company_2 ,
                FILENAME =
                    'C:\BBMTRIGGERS_TRIAL\BBM_ASPACE_NDF1.ndf' ,
                SIZE = 10MB ,
                MAXSIZE = 50 ,
                FILEGROWTH = 5% ) ,
                FILEGROUP BBM_USERS_1
                    (
                        NAME = BBM_Usuario1 ,
                        FILENAME =
                            'C:\BBMTRIGGERS_TRIAL\BBM_ASPACE_NDF2.ndf' ,
                        SIZE = 10MB ,
                        MAXSIZE = 50 ,
                        FILEGROWTH = 5% ) ,
                    (
                        NAME = BBM_USERS_EXTENSION ,
                        FILENAME =
                            'C:\BBMTRIGGERS_TRIAL\BBM_ASPACE_NDF3.ndf' ,
                        SIZE = 10MB ,
                        MAXSIZE = 50 ,
                        FILEGROWTH = 5% ) ,

                FILEGROUP BBM_EXPEDIENTS
                    (
                        NAME = BBM_Usuario2 ,
                        FILENAME =
                            'C:\BBMTRIGGERS_TRIAL\BBM_ASPACE_NDF4.ndf' ,
                        SIZE = 10MB ,
                        MAXSIZE = 50 ,
                        FILEGROWTH = 5% ) ,
                    (
                        NAME = BBM_EXPEDIENTS_EXTENSION ,
                        FILENAME =
                            'C:\BBMTRIGGERS_TRIAL\BBM_ASPACE_NDF5.ndf' ,
                        SIZE = 10MB ,
                        MAXSIZE = 50 ,
                        FILEGROWTH = 5% )

LOG ON
    (
        NAME = BBM_ASPACE_LOG ,
        FILENAME = 'C:\BBMTRIGGERS_TRIAL\BBM_ASPACE_LOGS.ldf' ,
        SIZE = 10MB ,
        MAXSIZE = 50 ,
        FILEGROWTH = 5% ) ;
GO

```



This PC > Local Disk (C:) > BBMTRIGGERS_TRIAL			
Name	Date modified	Type	Size
BBM_ASPACE_LOGS	03/03/2021 17:16	SQL Server Database Log	10,240 KB
BBM_ASPACE_MDF	03/03/2021 17:16	SQL Server Database File	10,240 KB
BBM_ASPACE_NDF1	03/03/2021 17:16	SQL Server Database File	10,240 KB
BBM_ASPACE_NDF2	03/03/2021 17:16	SQL Server Database File	10,240 KB
BBM_ASPACE_NDF3	03/03/2021 17:16	SQL Server Database File	10,240 KB
BBM_ASPACE_NDF4	03/03/2021 17:16	SQL Server Database File	10,240 KB
BBM_ASPACE_NDF5	03/03/2021 17:16	SQL Server Database File	10,240 KB

FILESTREAM

FILESTREAM permite a las aplicaciones basadas en SQL Server almacenar datos no estructurados, como documentos e imágenes, en el sistema de archivos. Usa la memoria caché del sistema NT para almacenar en memoria caché los datos de archivos. Esto ayuda a reducir cualquier efecto que los datos FILESTREAM podrían tener en el rendimiento de Motor de base de datos.

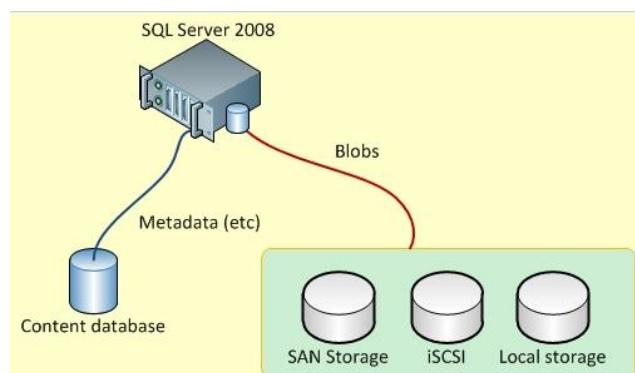
FILESTREAM integra Motor de base de datos de SQL Server con sistemas de archivos NTFS o ReFS almacenando datos de objetos binarios grandes (BLOB) varbinary(max) como archivos en el sistema de archivos. Transact-SQL pueden insertar, actualizar, consultar, buscar y realizar copias de seguridad de los datos FILESTREAM.

En SQL Server, los BLOB pueden ser datos de varbinary(max) estándar que almacena los datos en tablas u objetos FILESTREAM varbinary(max) que almacenan los datos en el sistema de archivos. El tamaño y el uso de los datos determinan si debería usar el almacenamiento de base de datos o el almacenamiento del sistema de archivos. Si las condiciones siguientes son verdaderas, debería pensar en usar FILESTREAM:

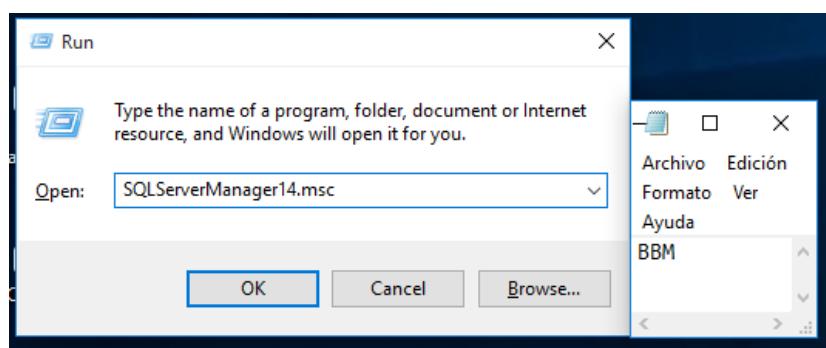
- Los objetos que se están almacenando son, por término medio, mayores de 1 MB.
- El acceso de lectura rápido es importante.
- Está desarrollando aplicaciones que usan un nivel intermedio para la lógica de la aplicación.

Para objetos de menor tamaño, el almacenamiento de BLOB varbinary(max) en la base de datos a menudo proporciona un mejor rendimiento de la transmisión de datos.

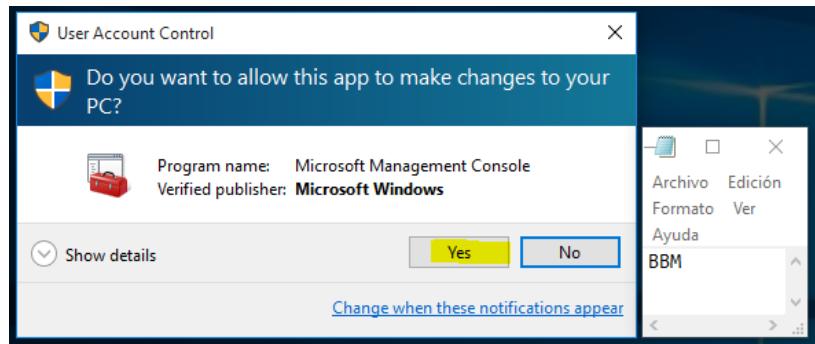
Como se verá en el script, el almacenamiento de *FILESTREAM* se implementa como una columna *varbinary(max)* en la que los datos están almacenados como *BLOB* en el sistema de archivos. Los tamaños de los *BLOB* están limitados solo por el tamaño del volumen del sistema de archivos. La limitación *varbinary(max)* estándar de tamaños de archivo de 2 GB no se aplica a *BLOB* que están almacenados en el sistema de archivos.

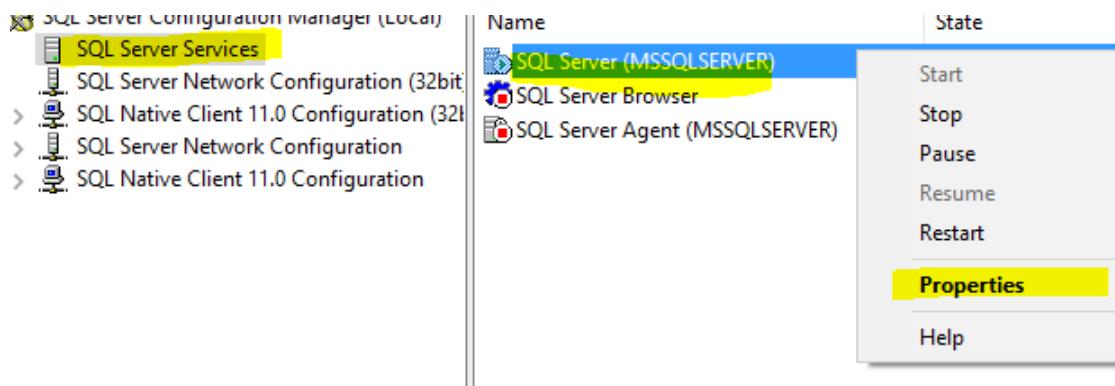


Antes de empezar a ejecutar *FILESTREAM* en el script hay que asegurarse de que está **habilitado como característica en SQL Server**. Hay dos maneras de configurarlo: una es durante la instalación, y otra (la que voy a hacer) es pulsando la tecla + R, escribimos **SQLServerManager14.msc**

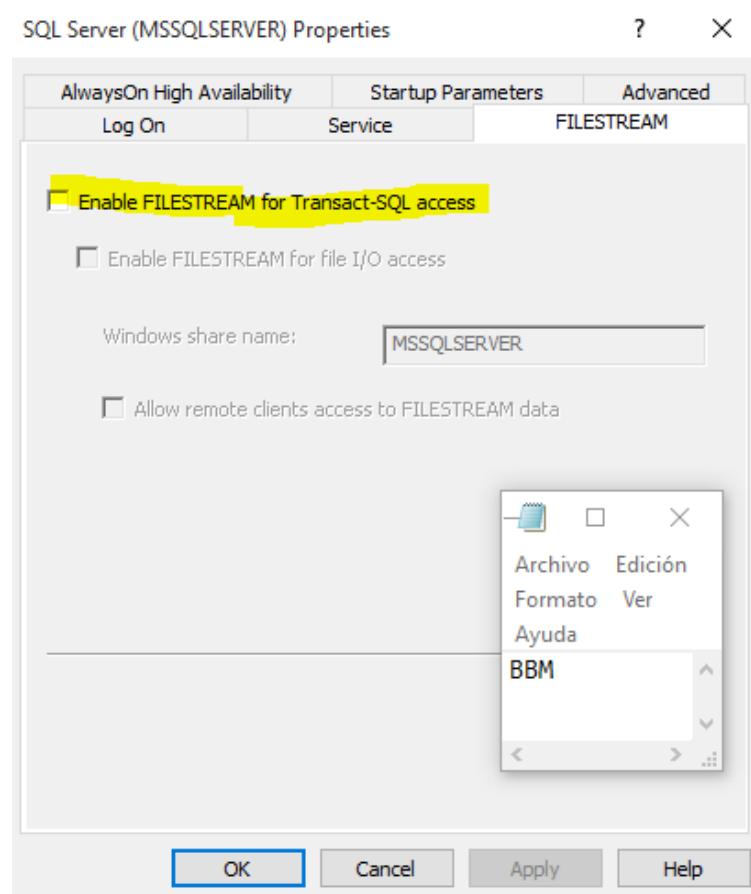


Aceptamos el cuadro que sale y accedemos a los servicios SQL Server. Una vez dentro, hacemos clic con el botón derecho sobre **SQL Server (MSSQLSERVER)** y le damos a **Properties**.

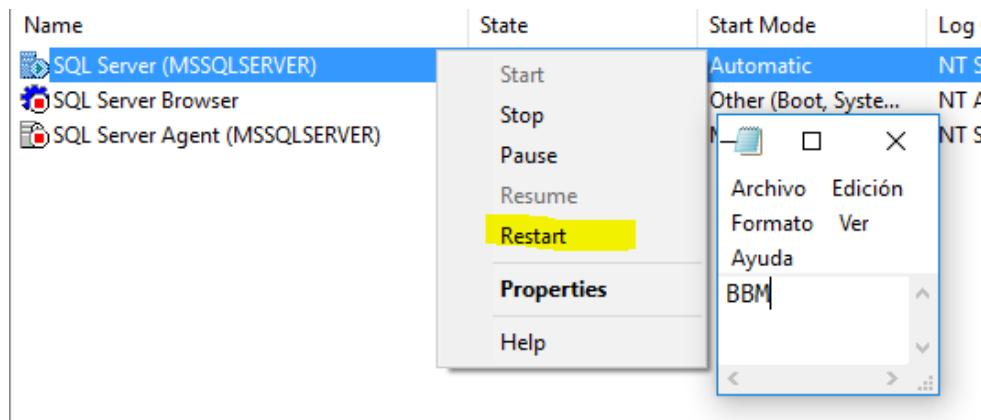




En las *Properties* vamos a la pestaña **FILESTREAM** y hacemos clic en el cuadro **Enable FILESTREAM for Transact-SQL Access**.



Con el SSMS cerrado, reiniciamos el servicio SQL Server (MSSQLSERVER)



Cuando iniciamos el SSMS, antes de poder empezar a utilizar lo que acabamos de configurar, hay que introducir el siguiente comando:

```
EXEC sp_configure filestream_access_level, 2
RECONFIGURE
GO
```

Donde a nivel de Base de datos mediante el **procedimiento almacenado `sp_configure`**, habilitamos el *FILESTREAM* con uno de los siguientes niveles (2 en nuestro caso):

- 0 = Deshabilitado. Este es el valor por defecto.
- 1 = Habilitado solo para acceso T-SQL.
- 2 = Habilitado solo para T-SQL y acceso local al sistema de ficheros.
- 3 = Habilitado para T-SQL, acceso local y remoto al sistema de ficheros.

Los datos de *FILESTREAM* deben estar almacenados en los grupos de archivos *FILESTREAM* (como el *FILEGROUP* que se creó arriba). Un grupo de archivos *FILESTREAM* es un grupo de archivos especial que contiene los directorios de sistema de archivos en lugar de los propios archivos. Estos directorios del sistema de archivos se denominan contenedores de datos. Los contenedores de datos son la interfaz entre el almacenamiento del Motor de base de datos y el almacenamiento del sistema de archivos.

Cuando use el almacenamiento *FILESTREAM*, piense en lo siguiente:

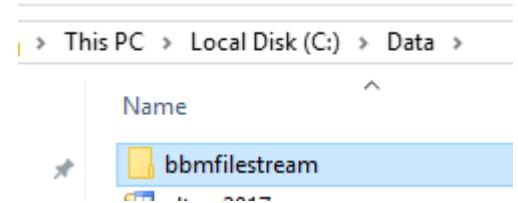
- Cuando una tabla contiene una columna *FILESTREAM*, cada fila debe tener un identificador de fila único distinto de NULL.
- Se pueden agregar varios contenedores de datos a un grupo de archivos *FILESTREAM*.
- Los contenedores de datos *FILESTREAM* no pueden estar anidados.

- Cuando se usan clústeres de conmutación por error, los grupos de archivos FILESTREAM deben estar en recursos de disco compartido.
- Los grupos de archivos FILESTREAM pueden estar en volúmenes comprimidos.

Para crear dicho grupo de archivos, utilizo esta parte del script donde altero la base de datos para generar los grupos de archivos de FILESTREAM:

```
-- AHORA A ESTA BASE LE CREO UN FILEGROUP PARA PODER METER TODO
USE TRIALBBM
GO

ALTER DATABASE TRIALBBM
    ADD FILEGROUP [PRIMARY_FILESTREAMBBM] -->
ORGANIZACIONES LÍJICAS DE ARCHIVOS FÍSICOS
    CONTAINS FILESTREAM
GO
ALTER DATABASE TRIALBBM
    ADD FILE (
        NAME = 'BBM_FILESTREAM',
        FILENAME = 'C:\Data\bbmfilestream'
    )
    TO FILEGROUP [PRIMARY_FILESTREAMBBM]
GO
```



Luego con el comando

```
SELECT      *      INTO      <nombre_base_futura>.<schema>.<tabla>      from
<nombre_base_origen>.<schema>.<tabla>
```

hice un volcado de las tablas del schema trial en una base de datos nueva llamada TRIALBBM. Para evitar un posible fallo, creé el schema previamente:

```
-- ME COLOCO EN LA BASE DE DATOS Y CREO UN ESQUEMA
USE TRIALBBM
GO
DROP SCHEMA IF EXISTS trial
GO
CREATE SCHEMA trial
GO
--CON LO CREADO ANTERIORMENTE, VUELCO LOS DATOS DE LAS TABLAS DE PRUEBAS
EN ESTA NUEVA TABLA PARA PODER TRABAJAR EN ELLA
Select * into TRIALBBM.trial.BBM_Expediente
BBM_ASPACE.trial.BBM_Expediente
from
Select * into TRIALBBM.trial.BBM_Session
BBM_ASPACE.trial.BBM_Sesion
from
Select * into TRIALBBM.trial.BBM_Comidas
BBM_ASPACE.trial.BBM_Comidas
from
Select * into TRIALBBM.trial.BBM_Enfermedad_Diagnosticada
BBM_ASPACE.trial.BBM_Enfermedad_Diagnosticada
from
Select * into TRIALBBM.trial.BBM_Tratamientos
BBM_ASPACE.trial.BBM_Tratamientos
from
```

```
Select * into TRIALBBM.trial.BBM_Usuario
      from
BBM_ASPACE.trial.BBM_Usuario
Select * into TRIALBBM.trial.BBM_Especialista
      from
BBM_ASPACE.trial.BBM_Especialista
```

A la base **TRIALBBM** a la cual le hicimos un volcado hago la suposición de que se me olvidó meter una tabla con las imágenes la cual iría asociada al usuario, y por tanto al expediente.

FILESTREAM no se habilita automáticamente al instalar o actualizar SQL Server y para utilizarlo, se debe crear o modificar una base de datos que contenga un tipo especial de grupo de archivos. Luego, debe crear o modificar una tabla de modo que contenga una columna **varbinary(max)** con el atributo **FILESTREAM**. Su estructura es la siguiente:

```
CREATE TABLE <nombre_tabla>(
    <campo_tabla> UNIQUEIDENTIFIER ROWGUIDCOL NOT NULL UNIQUE,
    <campo_tabla_FILESTREAM> VARBINARY (MAX) FILESTREAM
);
GO
```

Para especificar que una columna debería almacenar datos en el sistema de archivos, especifique el atributo **FILESTREAM** en una columna **varbinary(max)**. Esto hace que Motor de base de datos almacene todos los datos para esa columna en el sistema de archivos pero no en el archivo de base de datos.

Primero creo la tabla con un **identificador único ROWGUIDCOL NOT NULL UNIQUE**, y en el script, en la línea donde se mete el campo de la tabla, se añade la **palabra reservada FILESTREAM**.

```
-- COMO SE ME HA OLVIDADO METER LA TABLA QUE CONTIENE LAS IMAGENES
USE TRIALBBM
GO
DROP TABLE IF EXISTS BBM_Imágenes_Usuarios
GO
CREATE TABLE BBM_Imágenes_Usuarios (
    id UNIQUEIDENTIFIER ROWGUIDCOL NOT NULL UNIQUE,
    imagenesarchivo VARBINARY (MAX) FILESTREAM
);
GO
```

Creo una carpeta en la ruta C:\Fotos_BBMUSERS\ y meto tres fotos ahí dentro. Luego en la tabla hago una **inserción de valores** entre los que **Bulk** la ruta donde están las

fotos. Usando Transact-SQL, se puede insertar, actualizar y eliminar los datos de FILESTREAM:

- Se puede usar una operación de inserción para llenar previamente un campo FILESTREAM con un valor nulo, un valor vacío o un dato insertado relativamente corto..
- Al actualizar un campo FILESTREAM, modifica los datos de BLOB subyacentes en el sistema de archivos. Cuando un campo FILESTREAM está establecido en NULL, se eliminan los datos de BLOB asociados al campo. No se puede usar ningún **UPDATE**. **Write()**, para realizar actualizaciones parciales en los datos.
- Al eliminar una fila, o eliminar o truncar una tabla que contiene datos FILESTREAM, elimina los datos de BLOB subyacentes del sistema de archivos.

```
-- FOLDER C:\Fotos_BBMUSERS
INSERT INTO BBM_Imágenes_Usuarios(id, imagenesarchivo)
    SELECT NEWID(), BulkColumn
    FROM OPENROWSET(BULK 'C:\Fotos_BBMUSERS\USERGPP012.png',
SINGLE_BLOB) as f;
GO
INSERT INTO BBM_Imágenes_Usuarios(id, imagenesarchivo)
    SELECT NEWID(), BulkColumn
    FROM OPENROWSET(BULK 'C:\Fotos_BBMUSERS\USERVNA013.png', SINGLE_BLOB)
as f;
GO
INSERT INTO BBM_Imágenes_Usuarios(id, imagenesarchivo)
    SELECT NEWID(), BulkColumn
    FROM OPENROWSET(BULK 'C:\Fotos_BBMUSERS\USERVRI013.png', SINGLE_BLOB)
as f;
GO
SELECT *
FROM BBM_Imágenes_Usuarios;
GO
```

		Archivo
	id	Formato
1	6CE84714-7E04-4863-8C95-55421F6565E7	0x89504E470D0A1A0A0000000D494844520000023E000002...
2	846A76DD-A20F-4DE5-A273-94F26933B8DA	0x89504E470D0A1A0A0000000D494844520000023F000002...
3	4E83CC49-494B-4E26-9CF6-AC810C12D479	0x89504E470D0A1A0A0000000D494844520000023E000002...

Hago una consulta con una **Join** para ver el *FILESTREAM* creado:

```
--C:\Data\bbmfilestream
-- Open with PAINT

-- Filestream columns
SELECT SCHEMA_NAME(t.schema_id) AS [schema],
t.[name] AS [table],
c.[name] AS [column],
TYPE_NAME(c.user_type_id) AS [column_type]
FROM sys.columns c
JOIN sys.tables t ON c.object_id = t.object_id
```

```

WHERE t.filestream_data_space_id IS NOT NULL
    AND c.is_filestream = 1
ORDER BY 1, 2, 3;
-- Filestream files and filegroups
SELECT f.[name] AS [file_name],
    f.physical_name AS [file_path],
    fg.[name] AS [filegroup_name]
FROM sys.database_files f
JOIN sys.filegroups fg ON f.data_space_id = fg.data_space_id
WHERE f.[type] = 2
ORDER BY 1;
GO

```

	schema	table	column	column_type
1	dbo	BBM_Imágenes_Usuarios	imagenesarchivo	varbinary
	file_name	file_path	filegroup_name	
1	BBM_FILESTREAM	C:\Data\bbmfilestream	PRIMARYFILESTREAMBBM	

Borro la tabla borrando la columna y desasociando el Filestream:

```

ALTER TABLE [dbo].BBM_Imágenes_Usuarios DROP COLUMN
[imagenesarchivo]
GO
ALTER TABLE BBM_Imágenes_Usuarios SET (FILESTREAM_ON="NULL")
GO

ALTER DATABASE [TRIALBBM] REMOVE FILE BBM_FILESTREAM;
GO

--Msg 5042, Level 16, State 13, Line 134
--The file 'BBM_FILESTREAM' cannot be removed because it is not empty.

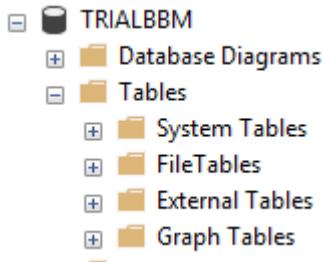
USE master
GO
-- The filegroup 'PRIMARYFILESTREAM' has been removed.
DROP DATABASE [TRIALBBM]
GO

```

NOTA: Dado que las operaciones de archivo son transaccionales, no puede eliminar ni cambiar el nombre de los archivos FILESTREAM a través del sistema de archivos.

FILETABLE

Una vez que nos conectamos a *SQL Server* y expandimos cualquier base de datos en SSMS, se obtienen las siguientes tablas:



- **System Tables:** SQL Server administra estas tablas internas de SQL Server
- **FileTables:** Las que voy a explicar
- **External Tables:** Las tablas externas son características nuevas en SQL Server 2017 en adelante
- **Graph Tables:** Graph tables are also a new feature in SQL Server 2017 onwards

SQL Server solo guarda metadatos en tablas de *FILESTREAM*. No se pueden controlar los objetos en el contenedor de *FILESTREAM* usando *SQL Server*, aunque proporciona coherencia a la transacción.

En el *SQL Server 2012* se introduce el concepto de *FILESTREAM*. Se pueden almacenar jerarquías de directorios en una base de datos. Proporciona compatibilidad con aplicaciones de Windows para los datos de archivo almacenados en *SQL Server*. *SQL FILETABLE* almacena los atributos de archivo y directorio junto con los datos de *FILESTREAM* en *FILETABLE*. El usuario puede acceder a los objetos desde el contenedor de forma similar a un archivo compartido. No tenemos que realizar cambios en las aplicaciones para acceder a estos datos. También puede pegar los archivos en el contenedor (directorio de host) y *SQL Server* inserta automáticamente metadatos en *FILETABLE*.

¿Qué quiere decir esto?, que ahora se pueden almacenar archivos y documentos en tablas especiales de *FILETABLES* denominadas *SQL Server* y tener acceso a ellos desde las aplicaciones Windows como si estuviesen almacenados en el sistema de archivos, sin efectuar cambios en las aplicaciones cliente.

Aprovecho toda la configuración del *FILESTREAM* y sólo ejecuto el reconfigure

```
EXEC sp_configure filestream_access_level, 2
RECONFIGURE --> PROVOCA LA EJECUCIÓN
GO
```

Creo primero la base de datos con los **Filegroup** (todo en una misma sentencia). Primero como siempre, controlando la existencia de la base de datos:

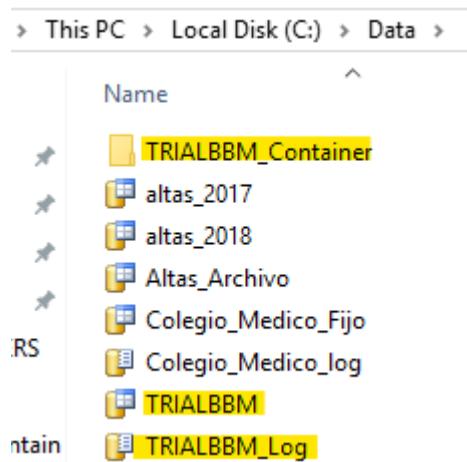
```
-FileTable

USE [master]
GO
DROP DATABASE IF EXISTS TRIALBBM
GO
```

```

CREATE DATABASE TRIALBBM
ON PRIMARY
(
    NAME = TRIALBBM_data,
    FILENAME = 'C:\Data\TRIALBBM.mdf' -->
MASTER DATA FILE
),
FILEGROUP FileStreamFG CONTAINS FILESTREAM
--> NOS LO GUARDA COMO NDF
(
    NAME = TRIALBBM,
    FILENAME =
'C:\Data\TRIALBBM_Container'
)
LOG ON
(
    NAME = TRIALBBM_Log,
    FILENAME = 'C:\Data\TRIALBBM_Log.ldf'
)
WITH FILESTREAM
(
    NON_TRANSACTED_ACCESS = FULL,
    DIRECTORY_NAME = 'TRIALBBMContainer'
);
GO

```



SQL Server proporciona una tabla de archivos especial, también denominada **FILETABLE**, para aplicaciones que requieren almacenamiento de archivos y directorios en la base de datos, con compatibilidad con la API de Windows y acceso no transaccional. La **FILETABLE** es una tabla de usuario especializada con un esquema predefinido que almacena los datos **FILESTREAM**, así como información de jerarquía de directorios y archivos e información de atributos de archivos.

Una **FILETABLE** proporciona la funcionalidad siguiente:

- Una **FILETABLE** representa una jerarquía de directorios y archivos. Almacena los datos relacionados con todos los nodos de esa jerarquía, tanto para los directorios como los archivos que contienen. Esta jerarquía se inicia en un directorio raíz que se especifica cuando se crea la **FILETABLE**.
- Cada fila de una **FILETABLE** representa un archivo o un directorio.
- Cada fila contiene los elementos siguientes. Para obtener más información acerca del esquema de una **FILETABLE**.
 - Una columna **file_stream** para los datos de secuencia y un identificador **stream_id** (GUID). (El valor de la columna **file_stream** es NULL para un directorio).

- Las dos columnas `path_locator` y `parent_path_locator` para representar y mantener el elemento actual (archivo o directorio) y la jerarquía de directorios.
- 10 atributos de archivo como fecha de creación y fecha de modificación que son útiles con las API de E/S de archivos.
- Una columna de tipo compatible con las búsquedas de texto completo y semántica en archivos y documentos.
- Una *FILETABLE* aplica ciertos desencadenadores y restricciones definidos por el sistema para mantener la semántica del espacio de nombres de archivo.
- Cuando se configura la base de datos para el acceso no transaccional, la jerarquía de archivos y directorios representada en *FILETABLE* se en el recurso compartido de *FILESTREAM* configurado para la instancia de *SQL SERVER*. De esta forma, se proporciona acceso al sistema de archivos para aplicaciones Windows.

Una manera de consultar las opciones de *filetable* es:

```
-- METADATA
-- Check the FileStream Options
SELECT DB_NAME(database_id),
non_transacted_access,
non_transacted_access_desc
FROM sys.database_filestream_options;
GO
-----
-- Another version
SELECT DB_NAME(database_id) as DatabaseName, non_transacted_access,
non_transacted_access_desc
FROM sys.database_filestream_options
where DB_NAME(database_id)='TRIALBBM';
GO
```

	DatabaseName	non_transacted_access	non_transacted_access_desc
1	TRIALBBM	2	FULL

----- SI VAMOS A LA BASE>TABLES>FILETABLES VEMOS QUE NOS LO CREA-----

Podemos tener las siguientes opciones para el acceso sin transacciones.

- **OFF**: No se permite el acceso no transaccional a *FILETABLES*
- **Read Only**: El acceso no transaccional a *FILETABLES* está permitido con fines de solo lectura

- Full: Se permite el acceso no transaccional a *FILETABLES* tanto para lectura como para escritura
- Especifica un directorio para el archivo SQL Server *FILETABLE*. Se necesita especificar el directorio sin la ruta del directorio. Este directorio actúa como una ruta raíz en la jerarquía *FILETABLE*.

Para este supuesto, utilizo la base de datos **TRIALBBM**, una base de datos que voy a usar como objeto de pruebas antes de incorporar a la base de datos final. Lo primero que hago es colocarme en la base de datos y controlar la existencia del *schema* que quiero crear. Una vez hecho esto, creo el *schema* trial.

```
-- ME COLOCO EN LA BASE DE DATOS Y CREO UN ESQUEMA
USE TRIALBBM
GO
DROP SCHEMA IF EXISTS trial
GO
CREATE SCHEMA trial
GO
```

Con lo creado anteriormente, vuelco los datos de las tablas de pruebas en esta nueva tabla para poder trabajar en ella:

```
--CON LO CREADO ANTERIORMENTE, VUELCO LOS DATOS DE LAS TABLAS DE PRUEBAS
EN ESTA NUEVA TABLA PARA PODER TRABAJAR EN ELLA
Select * into TRIALBBM.trial.BBM_Expediente      from
BBM_ASPACE.trial.BBM_Expediente
Select * into TRIALBBM.trial.BBM_Sesion           from
BBM_ASPACE.trial.BBM_Sesion
Select * into TRIALBBM.trial.BBM_Comidas          from
BBM_ASPACE.trial.BBM_Comidas
Select * into TRIALBBM.trial.BBM_Enfermedad_Diagnosticada   from
BBM_ASPACE.trial.BBM_Enfermedad_Diagnosticada
Select * into TRIALBBM.trial.BBM_Tratamientos     from
BBM_ASPACE.trial.BBM_Tratamientos
Select * into TRIALBBM.trial.BBM_Usuario           from
BBM_ASPACE.trial.BBM_Usuario
Select * into TRIALBBM.trial.BBM_Especialista     from
BBM_ASPACE.trial.BBM_Especialista
```

Al acabar creo la tabla con la estructura de siempre, pero con los condicionantes de que sea *FILETABLE* y que contenga un directorio que se llame **TRIALBBMContainer**:

```
-- Createee FileTable Table
USE TRIALBBM --> HABILITO FILETABLE
GO
DROP TABLE IF EXISTS BBM_FILETABLE_TRIAL
```

```

GO
CREATE TABLE BBM_FILETABLE_TRIAL --> CREO UNA TABLA ESPEC`ICA PARA
FILETABLE
AS FILETABLE
WITH
(
    FileTable_Directory = 'TRIALBBMContainer',
    FileTable_Collate_Filename = database_default
);
GO

```

Consulto la tabla para ver cómo queda:

```

SELECT *
FROM BBM_FILETABLE_TRIAL
GO

```

stream_id	file_stream	name	path_locator	parent_path_locator	file_type	cached_file_size	creation_time	last_write_time	last_access_time	is_directory	is_offline	is_hidden	is_READONLY	is_ARCHIVE	is_SYSTEM	is_TEMPORARY
-----------	-------------	------	--------------	---------------------	-----------	------------------	---------------	-----------------	------------------	--------------	------------	-----------	-------------	------------	-----------	--------------

Arrastro tres elementos a la carpeta que se crea y hago de nuevo una consulta:

```

-- Arrastro 3 objetos

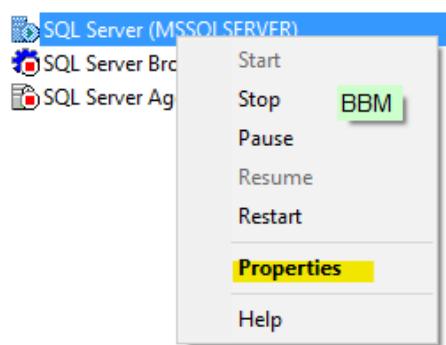
SELECT TOP (1000) [stream_id]
,[name]
FROM [TRIALBBM].[dbo].[BBM_FILETABLE_TRIAL]

-- stream_id name

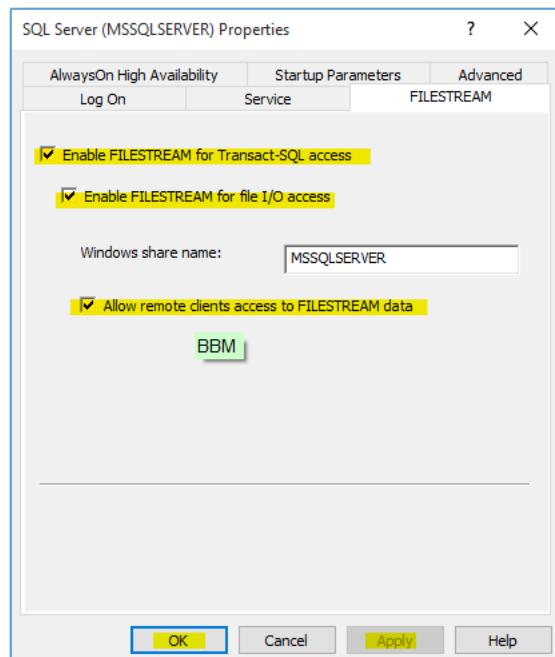
```

Como se puede ver, la consulta no funciona. Esto es debido a que la máquina está cerrada en un dominio, no permite hacer el Filetable en local, entonces hay que hacer lo siguiente:

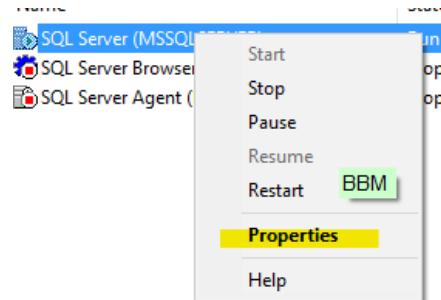
Voy al SQL Configuration Manager y sobre el servicio MSSQLSERVER le doy con el botón derecho y le doy a propiedades:



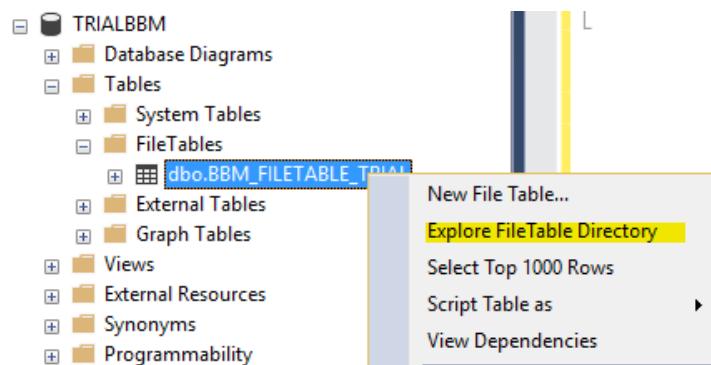
En la pestaña FileStream, le doy clic en las tres casillas, Aplico y le doy a OK:



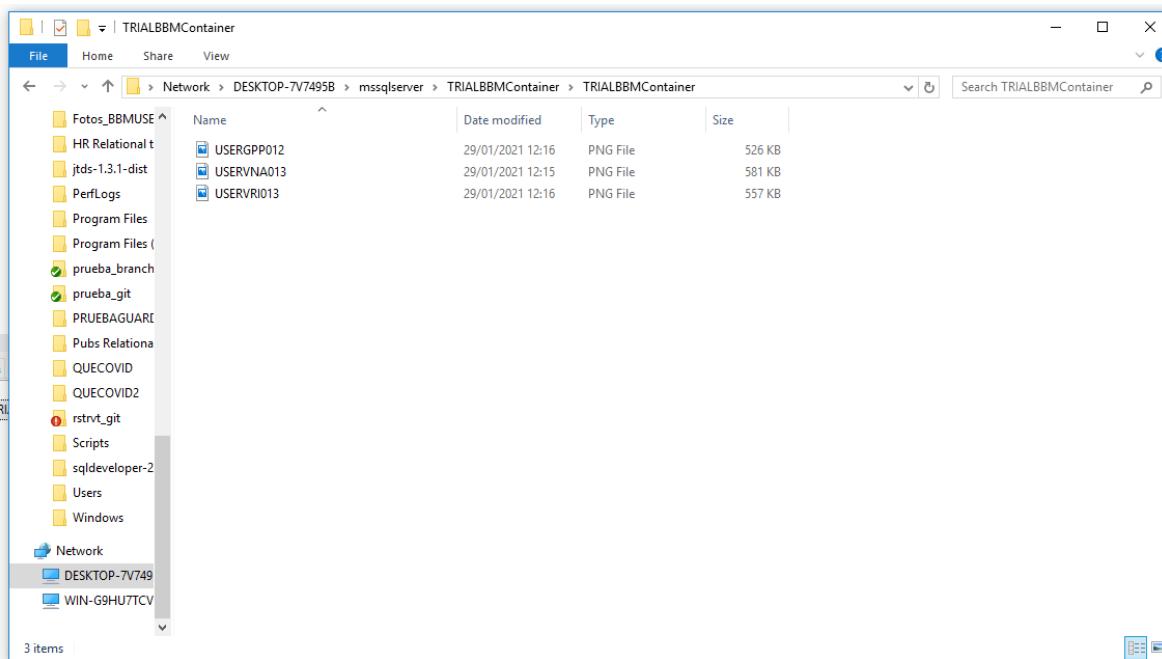
Reinicio el servicio:



Ahora la carpeta se crea en la red donde está el dominio, lo cual propicia que haya los permisos para poder insertar esas imágenes y que SQL Server las reconozca en el **Filetable**. Para llegar a ella, basta con darle a botón derecho sobre el **FILETABLE** y darle a **Explore FileTable Directory**:



Nos llevará a una carpeta donde suelto las tres fotos:



Hago la consulta:

```

SELECT TOP (1000) [stream_id]
, [name]
FROM [TRIALBBM].[dbo].[BBM_FILETABLE_TRIAL]

-- SUMMING UP

-- stream_id                                name
--B4608232-B146-EA11-9BCD-000C29A5C7F8    hiremenow.png
--B5608232-B146-EA11-9BCD-000C29A5C7F8    names.xls
--B7608232-B146-EA11-9BCD-000C29A5C7F8    Seguridad
Encerado.jpeg

SELECT * FROM sys.tables WHERE is_filetable = 1;
GO

```

BASES DE DATOS CONTENIDAS

PARTICIONES

Cuando se trabaja con las tablas, para mejorar la forma de almacenamiento y acceso a los datos. Es la opción de particionamiento de la tabla. Esta opción también ofrece poder trabajar y realizar ciertas labores de mantenimiento sobre conjuntos de datos, y no sobre toda la tabla.

El particionamiento de una tabla consiste en dividir una tabla grande en conjuntos más pequeños y manejables, sin tener que crear tablas nuevas. Estos conjuntos más pequeños se utilizan a la hora de administrar y mantener dicha tabla. Estas particiones son transparentes para el usuario final, que lanzará sus *queries* sin tener en cuenta cómo están los datos almacenados.

Hay dos tipos de particionamiento: particionamiento vertical y particionamiento horizontal.

El **particionamiento vertical** de tablas es principalmente usado para incrementar el desempeño de SQL Server especialmente en casos cuando una consulta retorna todas las columnas de una tabla que contiene un número de columnas de texto muy amplio o BLOB. En este caso, para reducir los tiempos de acceso, las columnas BLOB pueden ser divididas a su propia tabla. Otro ejemplo es restringir el acceso a datos sensibles, por ejemplo contraseñas, información salarial, etc. La partición vertical divide una tabla en dos o más tablas que contienen diferentes columnas.

El **particionamiento horizontal** divide una tabla en múltiples tablas que contienen el mismo número de columnas, pero menos filas. Por ejemplo, si una tabla contiene un gran número de filas que representan reportes mensuales podría ser particionada horizontalmente **en tablas por años** (el ejemplo que voy a usar), con cada tabla representando todos los reportes para un año específico. De esta manera las consultas que requieren datos para un año específico sólo referenciarán la tabla apropiada. Las tablas deberían ser particionadas en una manera que las consultas referencian tan pocas tablas como sea posible.

Las tablas son **particionadas horizontalmente basadas en una columna que será usada para particionar y los rangos asociados a cada partición**. La columna de particionamiento es usualmente una columna de **fecha**, pero todos los tipos de datos que son válidos para usarse como columnas de índice pueden ser usados como columna de partición, excepto columnas **timestamp**. Los siguientes tipos de datos no pueden ser especificados: **ntext, text, image, xml, varchar(max), nvarchar(max), o varbinary(max)**, el tipo definido por el usuario Microsoft .NET Framework *common language runtime (CLR)*, columnas de tipo de datos de alias.

Hay dos enfoques diferentes que se pueden usar para lograr la partición de la tabla. El primero es **crear una nueva tabla particionada y simplemente copiar los datos desde su tabla existente en la nueva tabla y renombrarla**. El segundo enfoque es **particionar una tabla existente reconstruyendo o creando un índice agrupado en la tabla**.

Empezamos creando una base de datos, como siempre contolando primero su existencia y posteriormente añadiendo los archivos mdf y ndf.

```
USE master
GO
DROP DATABASE IF EXISTS TRIALBBM
GO
-- AL DIVIDIRLO POR FECHAS, PUEDO DIVIDIRLO POR RANGOS

CREATE DATABASE [TRIALBBM]
ON PRIMARY ( NAME = 'TRIALBBM',
              FILENAME = 'C:\Data\TRIALBBM_Fijo.mdf'
              ,
              SIZE = 15360KB , MAXSIZE = UNLIMITED,
              FILEGROWTH = 0 )
LOG ON ( NAME = 'TRIALBBM_log',
          FILENAME = 'C:\Data\TRIALBBM_log.ldf'
          ,
          SIZE = 10176KB , MAXSIZE = 2048GB , FILEGROWTH = 10%)
GO
```

Como en este ejemplo lo que quiero hacer es una serie de **Filegroups** para dividir por años el trabajo que se va a desempeñar. Me pongo en la base que voy a crear y hago

ALTER DATABASE <base_de_datos> ADD FILEGROUP <nombre_filegroup>.

```
USE TRIALBBM
GO

ALTER DATABASE [TRIALBBM] ADD FILEGROUP [BBMFILEGROUP_Archivo]
GO
ALTER DATABASE [TRIALBBM] ADD FILEGROUP [BBMFILEGROUP_2016]
GO
ALTER DATABASE [TRIALBBM] ADD FILEGROUP [BBMFILEGROUP_2017]
GO
ALTER DATABASE [TRIALBBM] ADD FILEGROUP [BBMFILEGROUP_2018]
GO

select * from sys.filegroups
GO
```

	name	data_space_id	type	type_desc	is_default	is_system	filegroup_guid	log_filegroup_id	is_read_only	is_autogrow_all_files
1	PRIMARY	1	FG	ROWS_FILEGROUP	1	0	NULL	NULL	0	0
2	BBMFILEGROUP_Archivo	2	FG	ROWS_FILEGROUP	0	0	33015628-9F5F-48C3-98EC-2D08AC25CCA3	NULL	0	0
3	BBMFILEGROUP_2016	3	FG	ROWS_FILEGROUP	0	0	6F55F83A-F206-45B7-911F-4DD6BE750686	NULL	0	0
4	BBMFILEGROUP_2017	4	FG	ROWS_FILEGROUP	0	0	312D3834-08EC-43BA-AD7C-7E774646C34E	NULL	0	0
5	BBMFILEGROUP_2018	5	FG	ROWS_FILEGROUP	0	0	C83431D7-827C-4C38-BD5E-5629F0F1F0B5	NULL	0	0

Una vez hecho eso, creo los diferentes Filegroups que van a conformar la prueba haciendo el comando **ALTER DATABASE <nombre_base> ADD FILE [archivo] TO FILEGROUP <nombre_filegroup>**.

```
ALTER DATABASE [TRIALBBM] ADD FILE ( NAME = 'Nuevas_Altas_BBM' , FILENAME
= 'C:\Data\Nuevas_Altas_BBM.ndf' , SIZE = 5MB , MAXSIZE = 100MB , FILEGROWTH
= 2MB ) TO FILEGROUP [BBMFILEGROUP_Archivo]
GO
```

```

ALTER DATABASE [TRIALBBM] ADD FILE ( NAME = 'ARCHIVO_2016', FILENAME =
'c:\Data\ARCHIVO_2016.ndf', SIZE = 5MB, MAXSIZE = 100MB, FILEGROWTH = 2MB
) TO FILEGROUP [BBMFILEGROUP_2016]
GO
ALTER DATABASE [TRIALBBM] ADD FILE ( NAME = 'ARCHIVO_2017', FILENAME =
'c:\Data\ARCHIVO_2017.ndf', SIZE = 5MB, MAXSIZE = 100MB, FILEGROWTH = 2MB
) TO FILEGROUP [BBMFILEGROUP_2017]
GO
ALTER DATABASE [TRIALBBM] ADD FILE ( NAME = 'ARCHIVO_2018', FILENAME =
'c:\Data\ARCHIVO_2018.ndf', SIZE = 5MB, MAXSIZE = 100MB, FILEGROWTH = 2MB
) TO FILEGROUP [BBMFILEGROUP_2018]
GO
select * from sys.filegroups
GO

```

	name	data_space_id	type	type_desc	is_default	is_system	filegroup_guid	log_filegroup_id	is_read_only	is_autogrow_all_files
1	PRIMARY	1	FG	ROWS_FILEGROUP	1	0	NULL	NULL	0	0
2	BBMFILEGROUP_Archivo	2	FG	ROWS_FILEGROUP	0	0	33015628-9F5F-48C3-98EC-2D08AC25CCA3	NULL	0	0
3	BBMFILEGROUP_2016	3	FG	ROWS_FILEGROUP	0	0	6F55F83A-F206-45B7-911F-4DD6BE750686	NULL	0	0
4	BBMFILEGROUP_2017	4	FG	ROWS_FILEGROUP	0	0	312D3834-08EC-43BA-AD7C-7E774646C34E	NULL	0	0
5	BBMFILEGROUP_2018	5	FG	ROWS_FILEGROUP	0	0	C83431D7-827C-4C38-BD5E-5629F01F0B5	NULL	0	0

```

select * from sys.database_files
GO

```

file_id	file_guid	type	type_desc	data_space_id	name	physical_name	state	state_desc	size	max_size	growth	is_media_read_only	is_read_only	is_sparse	is_perc
1	B78E9FA7-8605-42CA-A5F1-7B655EC376EF	0	ROWS	1	TRIALBBM	C:\Data\TRIALBBM_Fijo.mdf	0	ONLINE	1920	-1	0	0	0	0	
2	4CD54D84-DB1C-4915-B22F-8C7E34068788	1	LOG	0	TRIALBBM_log	C:\Data\TRIALBBM_Log.ldf	0	ONLINE	1272	268435456	10	0	0	1	
3	B8AAFA23-BE9D-4569-8863-06DD76A80F00	0	ROWS	2	Nuevas_Altas_BBM	c:\Data\Nuevas_Altas_BBM.mdf	0	ONLINE	640	12800	256	0	0	0	
4	532C44A7-F37C-4B8F-A379-3FE260CC6562	0	ROWS	3	ARCHIVO_2016	c:\Data\ARCHIVO_2016.ndf	0	ONLINE	640	12800	256	0	0	0	
5	85A62AAB-EDD1-4079-BB18-593545137A7F	0	ROWS	4	ARCHIVO_2017	c:\Data\ARCHIVO_2017.ndf	0	ONLINE	640	12800	256	0	0	0	
6	C803A40E-CBA3-465F-BAC1-1ADBC3E9E0B5	0	ROWS	5	ARCHIVO_2018	c:\Data\ARCHIVO_2018.ndf	0	ONLINE	640	12800	256	0	0	0	

Ahora creo un schema al que le voy a volcar datos:

```

-- ME COLOCO EN LA BASE DE DATOS Y CREO UN ESQUEMA
USE TRIALBBM
GO
DROP SCHEMA IF EXISTS trial
GO
CREATE SCHEMA trial
GO

--CON LO CREADO ANTERIORMENTE, VUELCO LOS DATOS DE LAS TABLAS DE PRUEBAS
EN ESTA NUEVA TABLA PARA PODER TRABAJAR EN ELLA
Select * into TRIALBBM.trial.BBM_Expediente
    from
BBM_ASPACE.trial.BBM_Expediente
Select * into TRIALBBM.trial.BBM_Sesion
    from
BBM_ASPACE.trial.BBM_Sesion
Select * into TRIALBBM.trial.BBM_Comidas
    from
BBM_ASPACE.trial.BBM_Comidas
Select * into TRIALBBM.trial.BBM_Enfermedad_Diagnosticada
    from
BBM_ASPACE.trial.BBM_Enfermedad_Diagnosticada
Select * into TRIALBBM.trial.BBM_Tratamientos
    from
BBM_ASPACE.trial.BBM_Tratamientos
Select * into TRIALBBM.trial.BBM_Usuario
    from
BBM_ASPACE.trial.BBM_Usuario

```

```
Select * into TRIALBBM.trial.BBM_Especialista
      from
BBM_ASPACE.trial.BBM_Especialista
```

¿Qué es lo que hace que se particione una tabla en *SQL Server*? Lo primero que se necesita es definir una función y un *schema*.

Para crear la función de particionado, debemos recurrir a la cláusula **CREATE PARTITION FUNCTION**. En ella vamos a especificar 3 cosas:

- El tipo de datos de entrada
- Los valores del umbral
- Hacia donde caen los valores del umbral en los valores límite (RIGHT o LEFT)

```
-- PARTITION FUNCTION
```

```
CREATE PARTITION FUNCTION PART_FUNCTION_BBM (datetime)
AS RANGE RIGHT
    FOR VALUES ('2016-01-01','2017-01-01')
GO
```

De momento tenemos una función de particionado, pero que no aplica a nada. Lo siguiente, por tanto, será crear el **esquema de particionado** al que asignar dicha función de particionado.

Un esquema de particionado sirve para mapear las particiones a los *Filegroups* que deseemos. Para crear el esquema de particionado, debemos recurrir a la cláusula **CREATE PARTITION SCHEME**. En dicha cláusula vamos a especificar 2 cosas únicamente:

- La función de particionado que va a dirigir su salida a cada partición
- Los nombres de los *Filegroups* a los que cada partición irá mapeada.

```
CREATE PARTITION SCHEME PART_SCHEMA_BBM
AS PARTITION PART_FUNCTION_BBM
    TO
(BBMFILEGROUP_Archivo,BBMFILEGROUP_2016,BBMFILEGROUP_2017,BBMFILEGROUP_20
18)
GO
-- Partition scheme 'PART_SCHEMA_BBM' has been created successfully.
'BBMFILEGROUP_2018' is marked as the next used filegroup in partition
scheme 'PART_SCHEMA_BBM'.
```

En el caso de que el objeto ya exista, podemos optar por dos opciones:

- Recrear su índice clustered, pero esta vez de forma particionada
- Utilizar ALTER TABLE SWITCH para modificar los datos de la tabla para que solo tenga una única partición, a la que posteriormente iremos añadiendo otras.

```
ALTER TABLE trial.BBM_Usuario
  ADD fecha_alta datetime
  ON PART_SCHEMA_BBM -- partition scheme
    (fecha_alta) -- the column to apply the function within the
scheme
GO
```

Por último, ya solo queda crear el objeto particionado. Para ello se utiliza la sintaxis habitual, pero añadiendo al final la referencia a qué función de particionado (a la que proporcionaremos la columna por la que particionar) por la que se particionará.

El objeto que se va a crear necesita dos requerimientos adicionales:

- La localización de almacenamiento está dada por las particiones de *schema* (con el nombre de las particiones que se usarán para el particionado)
- La tabla debe tener un índice agrupado (normalmente la PK) el cual incluye la columna para ser usada para el particionado

Controlo la existencia de la tabla y la creo

```
DROP TABLE IF EXISTS trial.BBM_Usuario
GO
CREATE TABLE [trial].[BBM_Usuario](
  Usuario_ID INT NOT NULL IDENTITY,
  Nombre_1 varchar(20) NULL,
  Apellido_1 varchar(20) NULL,
  DNI varchar(9) NOT NULL,
  fecha_alta datetime)
  ON PART_SCHEMA_BBM -- partition scheme
    (fecha_alta) -- the column to apply the function within the
scheme
GO
```

Inserto valores y hago tres consultas:

```
INSERT INTO trial.BBM_Usuario (Nombre_1, Apellido_1, DNI, fecha_alta)
  Values ('Carlos','Ariza','12345678A','2015-01-01'),
('Victorino','Navas','12345678B','2013-05-05'),
('Debora','Moran','12345678C','2014-08-11')
GO
```

```
SELECT * FROM trial.BBM_Usuario
GO
```

	Usuario_ID	Nombre_1	Apellido_1	DNI	fecha_alta
1	1	Carlos	Ariza	12345678A	2015-01-01 00:00:00.000
2	2	Victorino	Navas	12345678B	2013-05-05 00:00:00.000
3	3	Debora	Moran	12345678C	2014-08-11 00:00:00.000

```
-----
```

```
SELECT *, $Partition.PART_FUNCTION_BBM(fecha_alta) AS Partition
FROM trial.BBM_Usuario
GO
```

	Usuario_ID	Nombre_1	Apellido_1	DNI	fecha_alta	Partition
1	1	Carlos	Ariza	12345678A	2015-01-01 00:00:00.000	1
2	2	Victorino	Navas	12345678B	2013-05-05 00:00:00.000	1
3	3	Debora	Moran	12345678C	2014-08-11 00:00:00.000	1

```
-- partition function
select name, create_date, value from sys.partition_functions f
inner join sys.partition_range_values rv
on f.function_id=rv.function_id
where f.name = 'PART_FUNCTION_BBM'
go
```

	name	create_date	value
1	PART_FUNCTION_BBM	2021-03-04 17:42:52.457	2016-01-01 00:00:00.000
2	PART_FUNCTION_BBM	2021-03-04 17:42:52.457	2017-01-01 00:00:00.000

```
DECLARE @TableName NVARCHAR(200) = N'trial.BBM_Usuario'
SELECT SCHEMA_NAME(o.schema_id) + '.' + OBJECT_NAME(i.object_id) AS [object]
, p.partition_number AS [p#] , BBMFILEGROUP.name AS [filegroup]
, p.rows , au.total_pages AS pages , CASE boundary_value_on_right WHEN 1
THEN 'less than' ELSE 'less than or equal to' END AS comparison ,
rv.value , CONVERT (VARCHAR(6), CONVERT (INT, SUBSTRING (au.first_page,
6, 1) + SUBSTRING (au.first_page, 5, 1))) + ':' + CONVERT (VARCHAR(20),
CONVERT (INT, SUBSTRING (au.first_page, 4, 1)) + SUBSTRING (au.first_page,
3, 1) + SUBSTRING (au.first_page, 2, 1) + SUBSTRING (au.first_page, 1,
1))) AS first_page FROM sys.partitions p INNER JOIN sys.indexes i ON
p.object_id = i.object_id AND p.index_id = i.index_id INNER JOIN
sys.objects o
ON p.object_id = o.object_id INNER JOIN
sys.system_internals_allocation_units au ON p.partition_id =
au.container_id INNER JOIN sys.partition_schemes ps ON ps.data_space_id =
i.data_space_id INNER JOIN sys.partition_functions f ON f.function_id =
ps.function_id INNER JOIN sys.destination_data_spaces dds ON
```

```

dds.partition_scheme_id = ps.data_space_id AND dds.destination_id =
p.partition_number INNER JOIN sys.filegroups BBMFILEGROUP ON
dds.data_space_id = BBMFILEGROUP.data_space_id LEFT OUTER JOIN
sys.partition_range_values rv ON f.function_id = rv.function_id AND
p.partition_number = rv.boundary_id WHERE i.index_id < 2 AND o.object_id
= OBJECT_ID(@TableName);
GO

```

	object	p#	filegroup	rows	pages	comparison	value	first_page
1	trial.BBM_Usuario	1	BBMFILEGROUP_Archivo	3	9	less than	2016-01-01 00:00:00.000	3:8
2	trial.BBM_Usuario	2	BBMFILEGROUP_2016	0	0	less than	2017-01-01 00:00:00.000	0:0
3	trial.BBM_Usuario	3	BBMFILEGROUP_2017	0	0	less than	NULL	0:0

Hago otra inserción y se ve como va cambiando, ahora aparece una columna nueva llamada **colum_name**, donde se ven las inserciones que voy haciendo. En un siguiente ejemplo se verá como sube:

```

INSERT INTO trial.BBM_Usuario
VALUES ('Cesar','Medrano','12345678D','2016-06-23'),
('Julia','Cañete','12345678E','2016-02-03'),
('Noa','Ferrando','12345678F','2016-04-06')
GO

```

```

SELECT *, $Partition.PART_FUNCTION_BBM(fecha_alta)
FROM trial.BBM_Usuario
GO

```

	Usuario_ID	Nombre_1	Apellido_1	DNI	fecha_alta	(No column name)
1	1	Carlos	Ariza	12345678A	2015-01-01 00:00:00.000	1
2	2	Victorino	Navas	12345678B	2013-05-05 00:00:00.000	1
3	3	Debora	Moran	12345678C	2014-08-11 00:00:00.000	1
4	4	Cesar	Medrano	12345678D	2016-06-23 00:00:00.000	2
5	5	Julia	Cañete	12345678E	2016-02-03 00:00:00.000	2
6	6	Noa	Ferrando	12345678F	2016-04-06 00:00:00.000	2

```

select name, create_date, value from sys.partition_functions f
inner join sys.partition_range_values rv
on f.function_id=rv.function_id
where f.name = 'PART_FUNCTION_BBM'
go

```

	name	create_date	value
1	PART_FUNCTION_BBM	2021-03-04 17:42:52.457	2016-01-01 00:00:00.000
2	PART_FUNCTION_BBM	2021-03-04 17:42:52.457	2017-01-01 00:00:00.000

```

select p.partition_number, p.rows from sys.partitions p
inner join sys.tables t
on p.object_id=t.object_id and t.name = 'trial.BBM_Usuario'
go

```

```

partition_number    rows

DECLARE @TableName NVARCHAR(200) = N'trial.BBM_Usuario'
SELECT SCHEMA_NAME(o.schema_id) + '.' + OBJECT_NAME(i.object_id) AS [object]
, p.partition_number AS [p#] , BBMFILEGROUP.name AS [filegroup]
, p.rows , au.total_pages AS pages , CASE boundary_value_on_right WHEN 1
THEN 'less than' ELSE 'less than or equal to' END AS comparison ,
rv.value , CONVERT (VARCHAR(6), CONVERT (INT, SUBSTRING (au.first_page,
6, 1) + SUBSTRING (au.first_page, 5, 1))) + ':' + CONVERT (VARCHAR(20),
CONVERT (INT, SUBSTRING (au.first_page, 4, 1)) + SUBSTRING (au.first_page,
3, 1) + SUBSTRING (au.first_page, 2, 1) + SUBSTRING (au.first_page, 1,
1))) AS first_page FROM sys.partitions p INNER JOIN sys.indexes i ON
p.object_id = i.object_id AND p.index_id = i.index_id INNER JOIN
sys.objects o
ON p.object_id = o.object_id INNER JOIN
sys.system_internals_allocation_units au ON p.partition_id =
au.container_id INNER JOIN sys.partition_schemes ps ON ps.data_space_id =
i.data_space_id INNER JOIN sys.partition_functions f ON f.function_id =
ps.function_id INNER JOIN sys.destination_data_spaces dds ON
dds.partition_scheme_id = ps.data_space_id AND dds.destination_id =
p.partition_number INNER JOIN sys.filegroups BBMFILEGROUP ON
dds.data_space_id = BBMFILEGROUP.data_space_id LEFT OUTER JOIN
sys.partition_range_values rv ON f.function_id = rv.function_id AND
p.partition_number = rv.boundary_id WHERE i.index_id < 2 AND o.object_id
= OBJECT_ID(@TableName);
GO

```

	object	p#	filegroup	rows	pages	comparison	value	first_page
1	trial.BBM_Usuario	1	BBMFILEGROUP_Archivo	3	9	less than	2016-01-01 00:00:00.000	3:8
2	trial.BBM_Usuario	2	BBMFILEGROUP_2016	3	9	less than	2017-01-01 00:00:00.000	4:8
3	trial.BBM_Usuario	3	BBMFILEGROUP_2017	0	0	less than	NULL	0:0

Vuelvo a hacer otra inserción para que se vaya viendo lo que estoy haciendo:

```

INSERT INTO trial.BBM_Usuario
VALUES ('Kevin','Pascual','12345678H','2017-05-21'),
('Evangelina','Lamas','12345678I','2017-07-09'),
('Cesar','Lopez','12345678J','2017-09-12')
GO

```

```

SELECT *,$Partition.PART_FUNCTION_BBM(fecha_alta)
FROM trial.BBM_Usuario
GO

```

	Usuario_ID	Nombre_1	Apellido_1	DNI	fecha_alta	(No column name)
1	1	Carlos	Ariza	12345678A	2015-01-01 00:00:00.000	1
2	2	Victorino	Navas	12345678B	2013-05-05 00:00:00.000	1
3	3	Debora	Moran	12345678C	2014-08-11 00:00:00.000	1
4	4	Cesar	Medrano	12345678D	2016-06-23 00:00:00.000	2
5	5	Julia	Cañete	12345678E	2016-02-03 00:00:00.000	2
6	6	Noa	Ferrando	12345678F	2016-04-06 00:00:00.000	2
7	7	Kevin	Pascual	12345678H	2017-05-21 00:00:00.000	3
8	8	Evangelina	Lamas	12345678I	2017-07-09 00:00:00.000	3
9	9	Cesar	Lopez	12345678J	2017-09-12 00:00:00.000	3

```
select name, create_date, value from sys.partition_functions f
inner join sys.partition_range_values rv
on f.function_id=rv.function_id
where f.name = 'PART_FUNCTION_BBM'
go
```

	name	create_date	value
1	PART_FUNCTION_BBM	2021-03-04 17:42:52.457	2016-01-01 00:00:00.000
2	PART_FUNCTION_BBM	2021-03-04 17:42:52.457	2017-01-01 00:00:00.000

```
select p.partition_number, p.rows from sys.partitions p
inner join sys.tables t
on p.object_id=t.object_id and t.name = 'trial.BBM_Usuario'
GO
```

partition_number	rows

```
DECLARE @TableName NVARCHAR(200) = N'trial.BBM_Usuario'
SELECT SCHEMA_NAME(o.schema_id) + '.' + OBJECT_NAME(i.object_id) AS [object]
, p.partition_number AS [p#] , BBMFILEGROUP.name AS [filegroup]
, p.rows , au.total_pages AS pages , CASE boundary_value_on_right WHEN 1
THEN 'less than' ELSE 'less than or equal to' END AS comparison ,
rv.value , CONVERT (VARCHAR(6), CONVERT (INT, SUBSTRING (au.first_page,
6, 1) + SUBSTRING (au.first_page, 5, 1))) + ':' + CONVERT (VARCHAR(20),
CONVERT (INT, SUBSTRING (au.first_page, 4, 1) + SUBSTRING (au.first_page,
3, 1) + SUBSTRING (au.first_page, 2, 1) + SUBSTRING (au.first_page, 1,
1))) AS first_page FROM sys.partitions p INNER JOIN sys.indexes i ON
p.object_id = i.object_id AND p.index_id = i.index_id INNER JOIN
sys.objects o
ON p.object_id = o.object_id INNER JOIN
sys.system_internals_allocation_units au ON p.partition_id =
au.container_id INNER JOIN sys.partition_schemes ps ON ps.data_space_id =
i.data_space_id INNER JOIN sys.partition_functions f ON f.function_id =
ps.function_id INNER JOIN sys.destination_data_spaces dds ON
dds.partition_scheme_id = ps.data_space_id AND dds.destination_id =
p.partition_number INNER JOIN sys.filegroups BBMFILEGROUP ON
dds.data_space_id = BBMFILEGROUP.data_space_id LEFT OUTER JOIN
sys.partition_range_values rv ON f.function_id = rv.function_id AND
p.partition_number = rv.boundary_id WHERE i.index_id < 2 AND o.object_id
= OBJECT_ID(@TableName);
GO
```

	object	p#	filegroup	rows	pages	comparison	value	first_page
1	trial.BBM_Usuario	1	BBMFILEGROUP_Archivo	3	9	less than	2016-01-01 00:00:00.000	3:8
2	trial.BBM_Usuario	2	BBMFILEGROUP_2016	3	9	less than	2017-01-01 00:00:00.000	4:8
3	trial.BBM_Usuario	3	BBMFILEGROUP_2017	3	9	less than	NULL	5:8

NOTA: En este vemos como el tercer Filegroup se va llenando con fechas de 2017

Inserto fechas de 2018 y hago

```
INSERT INTO trial.BBM_Usuario
VALUES ('Davinia','Cantero','12345678K','2018-02-12'),
('Josu','Cordoba','12345678L','2018-01-23'),
('Pelayo','Florez','12345678M','2018-02-23')
GO
```

```
SELECT *, $Partition.PART_FUNCTION_BBM(fecha_alta) as PARTITION
FROM trial.BBM_Usuario
GO
```

	Usuario_ID	Nombre_1	Apellido_1	DNI	fecha_alta	PARTITION
1	1	Carlos	Ariza	12345678A	2015-01-01 00:00:00.000	1
2	2	Victorino	Navas	12345678B	2013-05-05 00:00:00.000	1
3	3	Debora	Moran	12345678C	2014-08-11 00:00:00.000	1
4	4	Cesar	Medrano	12345678D	2016-06-23 00:00:00.000	2
5	5	Julia	Cañete	12345678E	2016-02-03 00:00:00.000	2
6	6	Noa	Ferrando	12345678F	2016-04-06 00:00:00.000	2
7	7	Kevin	Pascual	12345678H	2017-05-21 00:00:00.000	3
8	8	Evangelina	Lamas	12345678I	2017-07-09 00:00:00.000	3
9	9	Cesar	Lopez	12345678J	2017-09-12 00:00:00.000	3
10	10	Davinia	Cantero	12345678K	2018-02-12 00:00:00.000	3
11	11	Josu	Cordoba	12345678L	2018-01-23 00:00:00.000	3
12	12	Pelayo	Florez	12345678M	2018-02-23 00:00:00.000	3

```
select name, create_date, value from sys.partition_functions f
inner join sys.partition_range_values rv
on f.function_id=rv.function_id
where f.name = 'PART_FUNCTION_BBM'
go
```

	name	create_date	value
1	PART_FUNCTION_BBM	2021-03-04 17:42:52.457	2016-01-01 00:00:00.000
2	PART_FUNCTION_BBM	2021-03-04 17:42:52.457	2017-01-01 00:00:00.000

```
select p.partition_number, p.rows from sys.partitions p
inner join sys.tables t
on p.object_id=t.object_id and t.name = 'trial.BBM_Usuario'
GO
```

	partition_number	rows

```

DECLARE @TableName NVARCHAR(200) = N'trial.BBM_Usuario'
SELECT SCHEMA_NAME(o.schema_id) + '.' + OBJECT_NAME(i.object_id) AS [object]
, p.partition_number AS [p#] , BBMFILEGROUP.name AS [filegroup]
, p.rows , au.total_pages AS pages , CASE boundary_value_on_right WHEN 1
THEN 'less than' ELSE 'less than or equal to' END AS comparison ,
rv.value , CONVERT (VARCHAR(6), CONVERT (INT, SUBSTRING (au.first_page,
6, 1) + SUBSTRING (au.first_page, 5, 1))) + ':' + CONVERT (VARCHAR(20),
CONVERT (INT, SUBSTRING (au.first_page, 4, 1)) + SUBSTRING (au.first_page,
3, 1) + SUBSTRING (au.first_page, 2, 1) + SUBSTRING (au.first_page, 1,
1))) AS first_page FROM sys.partitions p INNER JOIN sys.indexes i ON
p.object_id = i.object_id AND p.index_id = i.index_id INNER JOIN
sys.objects o
ON p.object_id = o.object_id INNER JOIN
sys.system_internals_allocation_units au ON p.partition_id =
au.container_id INNER JOIN sys.partition_schemes ps ON ps.data_space_id =
i.data_space_id INNER JOIN sys.partition_functions f ON f.function_id =
ps.function_id INNER JOIN sys.destination_data_spaces dds ON
dds.partition_scheme_id = ps.data_space_id AND dds.destination_id =
p.partition_number INNER JOIN sys.filegroups BBMFILEGROUP ON
dds.data_space_id = BBMFILEGROUP.data_space_id LEFT OUTER JOIN
sys.partition_range_values rv ON f.function_id = rv.function_id AND
p.partition_number = rv.boundary_id WHERE i.index_id < 2 AND o.object_id
= OBJECT_ID(@TableName);
GO

```

	object	p#	filegroup	rows	pages	comparison	value	first_page
1	trial.BBM_Usuario	1	BBMFILEGROUP_Archivo	3	9	less than	2016-01-01 00:00:00.000	3:8
2	trial.BBM_Usuario	2	BBMFILEGROUP_2016	3	9	less than	2017-01-01 00:00:00.000	4:8
3	trial.BBM_Usuario	3	BBMFILEGROUP_2017	6	9	less than	NULL	5:8

NOTA: Si nos fijamos, las rows de 2017 crecieron por insertar fechas superiores o iguales a ese año

Una de las ventajas de utilizar particionado es que existen operaciones exclusivamente optimizadas para estos escenarios, pensadas para mejorar la manejabilidad de objetos particionados. Esto quiere decir que las particiones no son estáticas y que, por tanto, podremos modificar el esquema de particionado amoldándolo a nuestras necesidades.

Las operaciones más comunes que podemos realizar con objetos particionados son:

- Split
- Merge
- Switch
- Partition

La operación **SPLIT** sirve para separar una partición en 2. Al hacerlo, hay que indicar qué Filegroup será el que se utilizará para los valores que caigan en la nueva partición. Al utilizar la operación **SPLIT** debes recordar los siguientes aspectos:

- El nuevo valor límite debe pertenecer al rango de valores de la función de particionado.
- El nuevo valor límite no puede ser un valor límite asignado actualmente en la función de particionado.
- Si al dividir una partición con **SPLIT**, se producen dos particiones con datos, SQL Server distribuirá respectivamente los datos a sus particiones, con lo que ocurrirá un crecimiento en el log de transacciones para albergar las operaciones **INSERT** y **DELETE** que internamente se producirán. Aproximadamente el crecimiento del log de transacciones será del tamaño de los datos a moverse a la nueva partición.

Hacemos un **ALTER** de la partición de función:

```
ALTER PARTITION FUNCTION PART_FUNCTION_BBM()
SPLIT RANGE ('2018-01-01');
GO
```

Vemos como ahora también distingue entre 2018

```
SELECT *, $Partition.PART_FUNCTION_BBM(fecha_alta) as PARTITION
FROM trial.BBM_Usuario
GO
```

	Usuario_ID	Nombre_1	Apellido_1	DNI	fecha_alta	PARTITION
1	1	Carlos	Ariza	12345678A	2015-01-01 00:00:00.000	1
2	2	Victorino	Navas	12345678B	2013-05-05 00:00:00.000	1
3	3	Debora	Moran	12345678C	2014-08-11 00:00:00.000	1
4	4	Cesar	Medrano	12345678D	2016-06-23 00:00:00.000	2
5	5	Julia	Cañete	12345678E	2016-02-03 00:00:00.000	2
6	6	Noa	Ferrando	12345678F	2016-04-06 00:00:00.000	2
7	7	Kevin	Pascual	12345678H	2017-05-21 00:00:00.000	3
8	8	Evangelina	Lamas	12345678I	2017-07-09 00:00:00.000	3
9	9	Cesar	Lopez	12345678J	2017-09-12 00:00:00.000	3
10	10	Davinia	Cantero	12345678K	2018-02-12 00:00:00.000	4
11	11	Josu	Cordoba	12345678L	2018-01-23 00:00:00.000	4
12	12	Pelayo	Florez	12345678M	2018-02-23 00:00:00.000	4

Ahora también se usa el *Filegroup 2018*:

```

DECLARE @TableName NVARCHAR(200) = N'trial.BBM_Usuario'
SELECT SCHEMA_NAME(o.schema_id) + '.' + OBJECT_NAME(i.object_id) AS
[object] , p.partition_number AS [p#] , BBMFILEGROUP.name AS [filegroup]
, p.rows , au.total_pages AS pages , CASE boundary_value_on_right WHEN 1
THEN 'less than' ELSE 'less than or equal to' END AS comparison ,
rv.value , CONVERT (VARCHAR(6), CONVERT (INT, SUBSTRING (au.first_page,
6, 1) + SUBSTRING (au.first_page, 5, 1))) + ':' + CONVERT (VARCHAR(20),
CONVERT (INT, SUBSTRING (au.first_page, 4, 1)) + SUBSTRING (au.first_page,
3, 1) + SUBSTRING (au.first_page, 2, 1) + SUBSTRING (au.first_page, 1,
1))) AS first_page FROM sys.partitions p INNER JOIN sys.indexes i ON
p.object_id = i.object_id AND p.index_id = i.index_id INNER JOIN
sys.objects o
ON p.object_id = o.object_id INNER JOIN
sys.system_internals_allocation_units au ON p.partition_id =
au.container_id INNER JOIN sys.partition_schemes ps ON ps.data_space_id =
i.data_space_id INNER JOIN sys.partition_functions f ON f.function_id =
ps.function_id INNER JOIN sys.destination_data_spaces dds ON
dds.partition_scheme_id = ps.data_space_id AND dds.destination_id =
p.partition_number INNER JOIN sys.filegroups BBMFILEGROUP ON
dds.data_space_id = BBMFILEGROUP.data_space_id LEFT OUTER JOIN
sys.partition_range_values rv ON f.function_id = rv.function_id AND
p.partition_number = rv.boundary_id WHERE i.index_id < 2 AND o.object_id =
= OBJECT_ID(@TableName);
GO

```

	object	p#	filegroup	rows	pages	comparison	value	first_page
1	trial.BBM_Usuario	1	BBMFILEGROUP_Archivo	3	9	less than	2016-01-01 00:00:00.000	3:8
2	trial.BBM_Usuario	2	BBMFILEGROUP_2016	3	9	less than	2017-01-01 00:00:00.000	4:8
3	trial.BBM_Usuario	3	BBMFILEGROUP_2017	3	9	less than	2018-01-01 00:00:00.000	5:8
4	trial.BBM_Usuario	4	BBMFILEGROUP_2018	3	9	less than	NULL	6:8

La operación **MERGE** nos va a permitir mezclar dos particiones solapadas en una única partición. La forma de realizar dicha acción consiste en eliminar uno de los valores límite

Como consideración importante, cabe destacar que las operaciones **SPLIT** y **MERGE**, interesa que sean realizadas sobre particiones vacías, ya que de lo contrario incurren en movimiento de datos, para albergar las filas a las particiones. Debido a ello, las operaciones de **SPLIT** y **MERGE** pueden suponer un aumento de espacio en el log de transacciones (donde las operaciones quedan reflejadas como **DELETE** e **INSERT**) y un aumento del tiempo de operación intrínseco al propio movimiento de los datos.

Ahora voy a meter todas las fechas anteriores a 2016 juntas en la misma partición con el comando **ALTER** sobre la partición de función.

```

ALTER PARTITION FUNCTION PART_FUNCTION_BBM ()
MERGE RANGE ('2016-01-01');
GO

```

Hago los **SELECT** para que se vea lo que digo:

```
SELECT *, $Partition.PART_FUNCTION_BBM(fecha_alta)
FROM trial.BBM_Usuario
GO
```

	Usuario_ID	Nombre_1	Apellido_1	DNI	fecha_alta	(No column name)
1	1	Carlos	Ariza	12345678A	2015-01-01 00:00:00.000	1
2	2	Victorino	Navas	12345678B	2013-05-05 00:00:00.000	1
3	3	Debora	Moran	12345678C	2014-08-11 00:00:00.000	1
4	4	Cesar	Medrano	12345678D	2016-06-23 00:00:00.000	1
5	5	Julia	Cañete	12345678E	2016-02-03 00:00:00.000	1
6	6	Noa	Ferrando	12345678F	2016-04-06 00:00:00.000	1
7	7	Kevin	Pascual	12345678H	2017-05-21 00:00:00.000	2
8	8	Evangelina	Lamas	12345678I	2017-07-09 00:00:00.000	2
9	9	Cesar	Lopez	12345678J	2017-09-12 00:00:00.000	2
10	10	Davinia	Cantero	12345678K	2018-02-12 00:00:00.000	3
11	11	Josu	Cordoba	12345678L	2018-01-23 00:00:00.000	3
12	12	Pelayo	Florez	12345678M	2018-02-23 00:00:00.000	3

```
DECLARE @TableName NVARCHAR(200) = N'trial.BBM_Usuario'
SELECT SCHEMA_NAME(o.schema_id) + '.' + OBJECT_NAME(i.object_id) AS [object] , p.partition_number AS [p#] , BBMFILEGROUP.name AS [filegroup] , p.rows , au.total_pages AS pages , CASE boundary_value_on_right WHEN 1 THEN 'less than' ELSE 'less than or equal to' END AS comparison , rv.value , CONVERT (VARCHAR(6), CONVERT (INT, SUBSTRING (au.first_page, 6, 1) + SUBSTRING (au.first_page, 5, 1))) + ':' + CONVERT (VARCHAR(20), CONVERT (INT, SUBSTRING (au.first_page, 4, 1)) + SUBSTRING (au.first_page, 3, 1) + SUBSTRING (au.first_page, 2, 1) + SUBSTRING (au.first_page, 1, 1))) AS first_page FROM sys.partitions p INNER JOIN sys.indexes i ON p.object_id = i.object_id AND p.index_id = i.index_id INNER JOIN sys.objects o ON p.object_id = o.object_id INNER JOIN sys.system_internals_allocation_units au ON p.partition_id = au.container_id INNER JOIN sys.partition_schemes ps ON ps.data_space_id = i.data_space_id INNER JOIN sys.partition_functions f ON f.function_id = ps.function_id INNER JOIN sys.destination_data_spaces dds ON dds.partition_scheme_id = ps.data_space_id AND dds.destination_id = p.partition_number INNER JOIN sys.filegroups BBMFILEGROUP ON dds.data_space_id = BBMFILEGROUP.data_space_id LEFT OUTER JOIN sys.partition_range_values rv ON f.function_id = rv.function_id AND p.partition_number = rv.boundary_id WHERE i.index_id < 2 AND o.object_id = OBJECT_ID(@TableName);
GO
```

object	p#	filegroup	rows	pages	comparison	value	first_page
trial.BBM_Usuario	1	BBMFILEGROUP_Archivo	6	9	less than	2017-01-01 00:00:00.000	3:8
trial.BBM_Usuario	2	BBMFILEGROUP_2017	3	9	less than	2018-01-01 00:00:00.000	5:8
trial.BBM_Usuario	3	BBMFILEGROUP_2018	3	9	less than	NULL	6:8

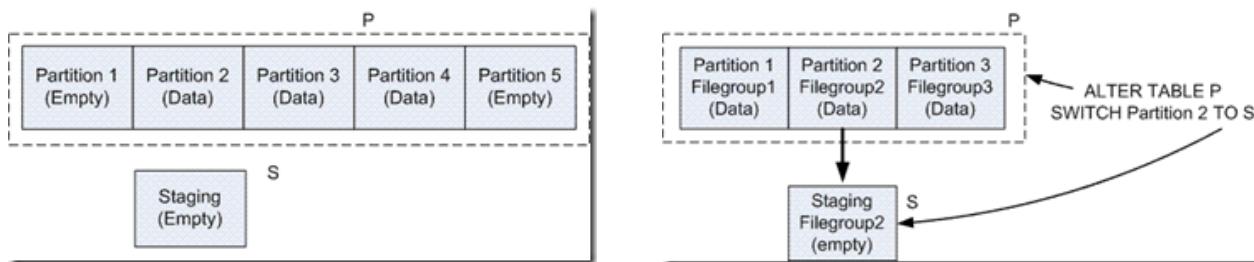
NOTA: El *Filegroup 2016* ha desaparecido (fusionado) con *Archivo*.

Siempre que sea posible, intentad que las operaciones SPLIT y MERGE se realicen sobre particiones vacías, ya que de esta forma se incurre solo en una modificación de esquema.

La operación **SWITCH** es una operación interesante para mover datos de forma rápida entre objetos. Es una operación que solo incurre en **modificación de metadatos** y que por tanto se realiza de forma instantánea (independientemente de cuantos datos existan en la partición).

Esta operación, es muy útil a la hora de extraer información de tablas a tablas históricas, puesto que de una forma rápida y sencilla se extrae el contenido de una partición de forma inmediata. El problema que resuelve esta operación es el que queda ilustrado en estas 2 imágenes:

- Tabla P, con datos en la partición 2 que quiero mover a una tabla S
- Se realiza la operación SWITCH y el resultado es que la tabla S se queda con los datos, mientras que la partición 2 de la tabla P se queda vacía.



Para esta prueba me cargo el *Filegroup 2016*:

```
USE master
GO
ALTER DATABASE [TRIALBBM] REMOVE FILE ARCHIVO_2016
go

ALTER DATABASE [TRIALBBM] REMOVE FILEGROUP BBMFILEGROUP_2016
go

--The file 'ARCHIVO_2016' has been removed.
--The filegroup 'BBMFILEGROUP_2016' has been removed.
```

Compruebo todos los cambios que se han sucedido al hacerlo:

```
select * from sys.filegroups
GO
```

	name	data_space_id	type	type_desc	is_default	is_system	filegroup_guid	log_filegroup_id	is_read_only	is_autogrow_all_files
1	PRIMARY	1	FG	ROWS_FILEGROUP	1	0	NULL	NULL	0	0

```
select * from sys.database_files
GO
```

	file_id	file_guid	type	type_desc	data_space_id	name	physical_name	state	state_desc	size	max_size	growth	is_media_read_only	is_read_only	is_sparse	is_percent_growth	is_name_reserved
1	1	NULL	0	ROWS	1	master	C:\Program Files\Microsoft SQL Server\MSSQL14.MSS...	0	ONLINE	688	-1	10	0	0	0	1	0
2	2	NULL	1	LOG	0	masterlog	C:\Program Files\Microsoft SQL Server\MSSQL14.MSS...	0	ONLINE	224	-1	10	0	0	0	1	0

```
USE TRIALBBM
go

SELECT *, $Partition.PART_FUNCTION_BBM(fecha_alta)
FROM trial.BBM_Usuario
GO
```

	Usuario_ID	Nombre_1	Apellido_1	DNI	fecha_alta	(No column name)
1	1	Carlos	Ariza	12345678A	2015-01-01 00:00:00.000	1
2	2	Victorino	Navas	12345678B	2013-05-05 00:00:00.000	1
3	3	Debora	Moran	12345678C	2014-08-11 00:00:00.000	1
4	4	Cesar	Medrano	12345678D	2016-06-23 00:00:00.000	1
5	5	Julia	Cañete	12345678E	2016-02-03 00:00:00.000	1
6	6	Noa	Ferrando	12345678F	2016-04-06 00:00:00.000	1
7	7	Kevin	Pascual	12345678H	2017-05-21 00:00:00.000	2
8	8	Evangelina	Lamas	12345678I	2017-07-09 00:00:00.000	2
9	9	Cesar	Lopez	12345678J	2017-09-12 00:00:00.000	2
10	10	Davinia	Cantero	12345678K	2018-02-12 00:00:00.000	3
11	11	Josu	Cordoba	12345678L	2018-01-23 00:00:00.000	3
12	12	Pelayo	Florez	12345678M	2018-02-23 00:00:00.000	3

```
DECLARE @TableName NVARCHAR(200) = N'trial.BBM_Usuario'
SELECT SCHEMA_NAME(o.schema_id) + '.' + OBJECT_NAME(i.object_id) AS [object]
, p.partition_number AS [p#] , BBMFILEGROUP.name AS [filegroup]
, p.rows , au.total_pages AS pages , CASE boundary_value_on_right WHEN 1
THEN 'less than' ELSE 'less than or equal to' END AS comparison ,
rv.value , CONVERT (VARCHAR(6), CONVERT (INT, SUBSTRING (au.first_page,
6, 1) + SUBSTRING (au.first_page, 5, 1))) + ':' + CONVERT (VARCHAR(20),
CONVERT (INT, SUBSTRING (au.first_page, 4, 1) + SUBSTRING (au.first_page,
3, 1) + SUBSTRING (au.first_page, 2, 1) + SUBSTRING (au.first_page, 1,
1))) AS first_page FROM sys.partitions p INNER JOIN sys.indexes i ON
p.object_id = i.object_id AND p.index_id = i.index_id INNER JOIN
sys.objects o
ON p.object_id = o.object_id INNER JOIN
sys.system_internals_allocation_units au ON p.partition_id =
au.container_id INNER JOIN sys.partition_schemes ps ON ps.data_space_id =
i.data_space_id INNER JOIN sys.partition_functions f ON f.function_id =
ps.function_id INNER JOIN sys.destination_data_spaces dds ON
dds.partition_scheme_id = ps.data_space_id AND dds.destination_id =
p.partition_number INNER JOIN sys.filegroups BBMFILEGROUP ON
dds.data_space_id = BBMFILEGROUP.data_space_id LEFT OUTER JOIN
sys.partition_range_values rv ON f.function_id = rv.function_id AND
p.partition_number = rv.boundary_id WHERE i.index_id < 2 AND o.object_id
= OBJECT_ID(@TableName);
GO
```

	object	p#	filegroup	rows	pages	comparison	value	first_page
1	trial.BBM_Usuario	1	BBMFILEGROUP_Archivo	6	9	less than	2017-01-01 00:00:00.000	3:8
2	trial.BBM_Usuario	2	BBMFILEGROUP_2017	3	9	less than	2018-01-01 00:00:00.000	5:8
3	trial.BBM_Usuario	3	BBMFILEGROUP_2018	3	9	less than	NULL	6:8

NOTA: Ha metido todo en Archivo

Creo la tabla **SWITCH_BBM** y hago una **ALTER** sobre la tabla **trial.BBM_Usuario** donde le indico que haga un switch de su primera partición a la nueva tabla que acabo de crear:

```
CREATE TABLE SWITCH_BBM
    (Usuario_ID INT NOT NULL IDENTITY,
     Nombre_1 varchar(20) NULL,
     Apellido_1 varchar(20) NULL,
     DNI varchar(9) NOT NULL,
     fecha_alta datetime)
ON BBMFILEGROUP_Archivo
go
```

```
ALTER TABLE trial.BBM_Usuario
    SWITCH Partition 1 to SWITCH_BBM
go
```

Compruebo todos los cambios:

```
select * from trial.BBM_Usuario
go
```

	Usuario_ID	Nombre_1	Apellido_1	DNI	fecha_alta
1	7	Kevin	Pascual	12345678H	2017-05-21 00:00:00.000
2	8	Evangelina	Lamas	12345678I	2017-07-09 00:00:00.000
3	9	Cesar	Lopez	12345678J	2017-09-12 00:00:00.000
4	10	Davinia	Cantero	12345678K	2018-02-12 00:00:00.000
5	11	Josu	Cordoba	12345678L	2018-01-23 00:00:00.000
6	12	Pelayo	Florez	12345678M	2018-02-23 00:00:00.000

```
select * from SWITCH_BBM
go
```

	Usuario_ID	Nombre_1	Apellido_1	DNI	fecha_alta
1	1	Carlos	Ariza	12345678A	2015-01-01 00:00:00.000
2	2	Victorino	Navas	12345678B	2013-05-05 00:00:00.000
3	3	Debora	Moran	12345678C	2014-08-11 00:00:00.000
4	4	Cesar	Medrano	12345678D	2016-06-23 00:00:00.000
5	5	Julia	Cañete	12345678E	2016-02-03 00:00:00.000
6	6	Noa	Ferrando	12345678F	2016-04-06 00:00:00.000

NOTA: Solo ha llevado lo que había en la Partición 1 de la tabla Usuario

```

DECLARE @TableName NVARCHAR(200) = N'trial.BBM_Usuario'
SELECT SCHEMA_NAME(o.schema_id) + '.' + OBJECT_NAME(i.object_id) AS [object]
, p.partition_number AS [p#] , BBMFILEGROUP.name AS [filegroup]
, p.rows , au.total_pages AS pages , CASE boundary_value_on_right WHEN 1
THEN 'less than' ELSE 'less than or equal to' END AS comparison ,
rv.value , CONVERT (VARCHAR(6), CONVERT (INT, SUBSTRING (au.first_page,
6, 1) + SUBSTRING (au.first_page, 5, 1))) + ':' + CONVERT (VARCHAR(20),
CONVERT (INT, SUBSTRING (au.first_page, 4, 1)) + SUBSTRING (au.first_page,
3, 1) + SUBSTRING (au.first_page, 2, 1) + SUBSTRING (au.first_page, 1,
1))) AS first_page FROM sys.partitions p INNER JOIN sys.indexes i ON
p.object_id = i.object_id AND p.index_id = i.index_id INNER JOIN
sys.objects o
ON p.object_id = o.object_id INNER JOIN
sys.system_internals_allocation_units au ON p.partition_id =
au.container_id INNER JOIN sys.partition_schemes ps ON ps.data_space_id =
i.data_space_id INNER JOIN sys.partition_functions f ON f.function_id =
ps.function_id INNER JOIN sys.destination_data_spaces dds ON
dds.partition_scheme_id = ps.data_space_id AND dds.destination_id =
p.partition_number INNER JOIN sys.filegroups BBMFILEGROUP ON
dds.data_space_id = BBMFILEGROUP.data_space_id LEFT OUTER JOIN
sys.partition_range_values rv ON f.function_id = rv.function_id AND
p.partition_number = rv.boundary_id WHERE i.index_id < 2 AND o.object_id
= OBJECT_ID(@Tabl
eName);
GO

```

	object	p#	filegroup	rows	pages	comparison	value	first_page
1	trial.BBM_Usuario	1	BBMFILEGROUP_Archivo	0	0	less than	2017-01-01 00:00:00.000	0:0
2	trial.BBM_Usuario	2	BBMFILEGROUP_2017	3	9	less than	2018-01-01 00:00:00.000	5:8
3	trial.BBM_Usuario	3	BBMFILEGROUP_2018	3	9	less than	NULL	6:8

NOTA: La partición 1 está vacía en la tabla *BBM_Usuario*

Los administradores de bases de datos de SQL Server prefieren usar la instrucción **TRUNCATE TABLE** en lugar de la instrucción **DELETE** porque es más rápida, mínimamente registrada y consume menos recursos del servidor. La desventaja de la declaración **TRUNCATE TABLE** es que elimina todas las filas de la tabla, ya que no hay una cláusula **WHERE** que se pueda agregar a la declaración para especificar los criterios de eliminación.

El comando es muy sencillo, se usa **TRUNCATE TABLE <nombre_tabla> WITH <número_de_particiones>**.

```

TRUNCATE TABLE trial.BBM_Usuario
WITH (PARTITIONS (3));
go
Compruebo que esto es cierto
select * from trial.BBM_Usuario
GO

```

	Usuario_ID	Nombre_1	Apellido_1	DNI	fecha_alta
1	7	Kevin	Pascual	12345678H	2017-05-21 00:00:00.000
2	8	Evangelina	Lamas	12345678I	2017-07-09 00:00:00.000
3	9	Cesar	Lopez	12345678J	2017-09-12 00:00:00.000

```

DECLARE @TableName NVARCHAR(200) = N'trial.BBM_Usuario'

SELECT SCHEMA_NAME(o.schema_id) + '.' + OBJECT_NAME(i.object_id) AS [object]
, p.partition_number AS [p#]
, BBMFILEGROUP.name AS [filegroup]
, p.rows
, au.total_pages AS pages
, CASE boundary_value_on_right WHEN 1 THEN 'less than' ELSE 'less than or equal to' END AS comparison
, rv.value
, CONVERT(VARCHAR(6), CONVERT(INT, SUBSTRING(au.first_page, 6, 1) + SUBSTRING(au.first_page, 5, 1))) + ':' + CONVERT(VARCHAR(20), CONVERT(INT, SUBSTRING(au.first_page, 4, 1)) + SUBSTRING(au.first_page, 3, 1) + SUBSTRING(au.first_page, 2, 1) + SUBSTRING(au.first_page, 1, 1))) AS first_page
FROM sys.partitions p INNER JOIN sys.indexes i ON p.object_id = i.object_id AND p.index_id = i.index_id
INNER JOIN sys.objects o
ON p.object_id = o.object_id
INNER JOIN sys.system_internals_allocation_units au ON p.partition_id = au.container_id
INNER JOIN sys.partition_schemes ps ON ps.data_space_id = i.data_space_id
INNER JOIN sys.partition_functions f ON f.function_id = ps.function_id
INNER JOIN sys.destination_data_spaces dds ON dds.partition_scheme_id = ps.data_space_id AND dds.destination_id = p.partition_number
INNER JOIN sys.filegroups BBMFILEGROUP ON dds.data_space_id = BBMFILEGROUP.data_space_id
LEFT OUTER JOIN sys.partition_range_values rv ON f.function_id = rv.function_id AND p.partition_number = rv.boundary_id WHERE i.index_id < 2 AND o.object_id = OBJECT_ID(@TableName);
GO

```

	object	p#	filegroup	rows	pages	comparison	value	first_page
1	trial.BBM_Usuario	1	BBMFILEGROUP_Archivo	0	0	less than	2017-01-01 00:00:00.000	0:0
2	trial.BBM_Usuario	2	BBMFILEGROUP_2017	3	9	less than	2018-01-01 00:00:00.000	5:8
3	trial.BBM_Usuario	3	BBMFILEGROUP_2018	0	0	less than	NULL	0:0

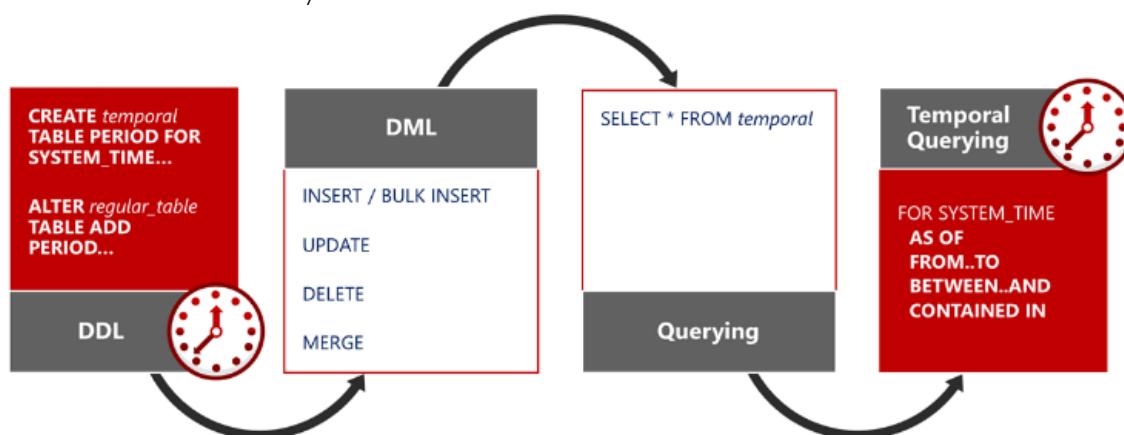
NOTA: Se ha borrado la *partición 3*, la del *Filegroup_2018*.

TABLAS TEMPORALES

Las tablas temporales nos habilitan la posibilidad de recuperar datos de cualquier punto en el pasado y por cada cambio de datos (Actualizar, Eliminar, Combinar). Con la tabla temporal, los usuarios pueden restaurar datos de cambios accidentales (actualizar/eliminar), así como verificar todos los cambios en los datos, etc.

Una tabla temporal se compone de dos partes (tablas):

- La primera es la tabla en la que se habilita el seguimiento de datos y se llama una tabla versión original del sistema, esta tabla contiene datos actuales (reales).
- La segunda parte es la tabla en la que se almacenan los cambios de datos que se realizaron (Actualizar, Eliminar, Combinar) en una tabla con versión del sistema y se llama tabla de historial.



Para esta prueba voy a usar el *schema trial* de la base de datos BBM_ASPACE. La activo y comienzo a crear una tabla temporal que me controle las reservas y el histórico de reservas de las aulas que se usan para los tratamientos de los pacientes, pues ahora con la pandemia, el uso de aulas está limitado y se necesita llevar un recuento de cómo se usa y cuándo:

```
USE BBM_ASPACE
GO
```

Cada tabla temporal debe tener dos columnas con un tipo de datos `datetime2`. Estas columnas son empleadas exclusivamente por el sistema para examinar el período de validez de cada fila siempre que se cambie una fila. Ambas columnas deben definirse como *NOT NULL*. Si no hay restricciones *NOT NULL* para las columnas `datetime2`, el sistema las mantendrá automáticamente como *NOT NULL*. Si las restricciones *NULL* se aplican a estas columnas, el sistema generará un error:

Una columna `datetime2` debe tener configurada la opción `GENERATED ALWAYS AS ROW START` y la segunda columna `datetime2` debe tener fijada la opción `GENERATED ALWAYS AS ROW END`:

- La columna `GENERATED ALWAYS AS ROW START` se refiere al momento en que los datos de la fila se volvieron actuales, básicamente en un `INSERT/UPDATE` del

registro en la tabla temporal del sistema con versión original, el sistema establecerá la hora UTC actual en función del reloj del sistema en el que el servidor SQL Server se ejecuta.

- La columna **GENERATED ALWAYS AS ROW END** representa el momento en que los datos de la fila ya no eran actuales. Esta columna indica la hora en que se producen los cambios **UPDATE/ DELETE**. El sistema establecerá la hora UTC actual.
- **Clave principal** así mismo una tabla temporal con versión del sistema debe contar con una clave primaria definida. Si la tabla no tiene una clave principal definida, aparecerá un error.

La tabla debe tener una cláusula **PERIOD FOR SYSTEM_TIME** (<startcol>, <endcol>) con dos columnas **datetime2** en ella. **PERIOD FOR SYSTEM_TIME** especifica el nombre de las columnas que el sistema manejará para registrar el período durante el cual un registro es válido.

La tabla también debe disponer de la cláusula **SYSTEM_VERSIONING**. **SYSTEM_VERSIONING** se usa para activar/desactivar las versiones del sistema para la tabla con versión del sistema.

```
DROP TABLE IF EXISTS trial.BBM_Espacio
GO

CREATE TABLE trial.BBM_Espacio
(
    ID_Espacio VARCHAR(20) PRIMARY KEY CLUSTERED,
    Reserva_Sala INTEGER,
    SysStartTime datetime2 GENERATED ALWAYS AS ROW START NOT NULL,
    SysEndTime datetime2 GENERATED ALWAYS AS ROW END NOT NULL,
    PERIOD FOR SYSTEM_TIME (SysStartTime,SysEndTime) )
WITH (SYSTEM_VERSIONING = ON (History_Table =
trial.reserva_sala_pendiente_desinfectar) -- histórico de salas
reservadas para su limpieza
)
GO
```

Después de ejecutar esta declaración, se producirá la tabla temporal de personas y se marcará como una tabla con versión del sistema. Dentro de la tabla de Personas se creará la tabla de Historia de Personas y se indicará como tabla de Historial:



Ahora hacemos una consulta de ambas tablas y vemos como está vacía:

```
SELECT * FROM [trial].[BBM_Espacio]
GO
```

ID_Espacio	Reserva_Sala	SysStartTime	SysEndTime
<code>SELECT * FROM [trial].[reserva_sala_pendiente_desinfectar]</code>			
<code>GO</code>			
ID_Espacio	Reserva_Sala	SysStartTime	SysEndTime

NOTA: Como se ve, ambas tienen los mismos campos.

Probamos a hacer una inserción de cinco salas. El valor booleano determina si están ocupadas (1) o vacías (2).

```
INSERT INTO [trial].[BBM_Espacio] ([ID_Espacio], Reserva_Sala)
VALUES ('Sala1','1'),
       ('Sala2','0'),
       ('Sala3','0'),
       ('Sala4','1'),
       ('Sala5','1')
GO
-- (5 rows affected)
```

Comprobamos y vemos que la una de las tablas está con los datos insertados y la otra vacía.

```
SELECT * FROM [trial].[BBM_Espacio]
GO
Sala1    1    2021-03-07 11:12:07.7256059 9999-12-31 23:59:59.9999999
Sala2    0    2021-03-07 11:12:07.7256059 9999-12-31 23:59:59.9999999
Sala3    0    2021-03-07 11:12:07.7256059 9999-12-31 23:59:59.9999999
Sala4    1    2021-03-07 11:12:07.7256059 9999-12-31 23:59:59.9999999
Sala5    1    2021-03-07 11:12:07.7256059 9999-12-31 23:59:59.9999999
SELECT * FROM [trial].[reserva_sala_pendiente_desinfectar]
GO
-- ID_Espacio  Reserva_Sala  SysStartTime  SysEndTime
```

Probamos a actualizar la *Sala1* y poner que ha quedado vacía.

```
UPDATE [trial].[BBM_Espacio]
SET Reserva_Sala = 0
WHERE ID_Espacio = 'Sala1'
GO
-- (1 row affected)
```

Como se ve en la consulta, la tabla *trial.BBM_Espacio* se actualiza con que la *Sala1* está vacía, y queda en la tabla *trial.reserva_sala_pendiente_desinfectar* que esa sala estuvo ocupada desde las 11:12:07 del 07-03-2021 hasta las 11:17:14 del 07-03-2021

```
SELECT * FROM [trial].[BBM_Espacio]
```

```
GO
```

```
/*ID_Espacio      Reserva_Sala      SysStartTime      SysEndTime
Sala1      0      2021-03-07 11:17:14.5542431 9999-12-31 23:59:59.9999999
Sala2      0      2021-03-07 11:12:07.7256059 9999-12-31 23:59:59.9999999
Sala3      0      2021-03-07 11:12:07.7256059 9999-12-31 23:59:59.9999999
Sala4      1      2021-03-07 11:12:07.7256059 9999-12-31 23:59:59.9999999
Sala5      1      2021-03-07 11:12:07.7256059 9999-12-31 23:59:59.9999999*/
```

```
SELECT * FROM [trial].[reserva_sala_pendiente_desinfectar]
GO
```

```
/*ID_Espacio      Reserva_Sala      SysStartTime      SysEndTime
Sala1      1      2021-03-07 11:12:07.7256059 2021-03-07 11:17:14.5542431*/
```

Probamos unas cuantas actualizaciones de salas más para que se vea un flujo continuado de entrada y salida de pacientes:

```
UPDATE [trial].[BBM_Espacio]
    SET Reserva_Sala = 1
    WHERE ID_Espacio = 'Sala3'
GO

--(1 row affected)
```

```
SELECT * FROM [trial].[BBM_Espacio]
GO

/*ID_Espacio      Reserva_Sala      SysStartTime      SysEndTime
Sala1      0      2021-03-07 11:17:14.5542431 9999-12-31 23:59:59.9999999
Sala2      0      2021-03-07 11:12:07.7256059 9999-12-31 23:59:59.9999999
Sala3      1      2021-03-07 11:19:59.1486039 9999-12-31 23:59:59.9999999
Sala4      1      2021-03-07 11:12:07.7256059 9999-12-31 23:59:59.9999999
Sala5      1      2021-03-07 11:12:07.7256059 9999-12-31 23:59:59.9999999*/
```

```
SELECT * FROM [trial].[reserva_sala_pendiente_desinfectar]
GO

/*ID_Espacio      Reserva_Sala      SysStartTime      SysEndTime
Sala1      1      2021-03-07 11:12:07.7256059 2021-03-07 11:17:14.5542431
Sala3      0      2021-03-07 11:12:07.7256059 2021-03-07 11:19:59.1486039*/
```

```
UPDATE [trial].[BBM_Espacio]
    SET Reserva_Sala = 0
    WHERE ID_Espacio = 'Sala5'
GO
```

```
--(1 row affected)
```

```
SELECT * FROM [trial].[BBM_Espacio]
GO

/*ID_Espacio      Reserva_Sala      SysStartTime      SysEndTime
Sala1      0      2021-03-07 11:17:14.5542431 9999-12-31 23:59:59.9999999
Sala2      0      2021-03-07 11:12:07.7256059 9999-12-31 23:59:59.9999999*/
```

```
Sala3    1    2021-03-07 11:19:59.1486039 9999-12-31 23:59:59.9999999
Sala4    1    2021-03-07 11:12:07.7256059 9999-12-31 23:59:59.9999999
Sala5    0    2021-03-07 11:21:25.6961863 9999-12-31 23:59:59.9999999*/
```

```
SELECT * FROM [trial].[reserva_sala_pendiente_desinfectar]
GO
```

```
/*ID_Espacio      Reserva_Sala      SysStartTime      SysEndTime
Sala1    1    2021-03-07 11:12:07.7256059 2021-03-07 11:17:14.5542431
Sala3    0    2021-03-07 11:12:07.7256059 2021-03-07 11:19:59.1486039
Sala5    1    2021-03-07 11:12:07.7256059 2021-03-07 11:21:25.6961863*/
```

Ahora hago una prueba: Imaginemos por un supuesto que una sala se cierra para desinfectar. Yo la borro de mi tabla:

```
DELETE FROM [trial].[BBM_Espacio]
WHERE ID_Espacio='Sala5'
GO

--(1 row affected)
```

Como se ve, la tabla ha desaparecido de la normal, pero en el histórico aún aparece con los movimientos que se han hecho:

```
SELECT * FROM [trial].[BBM_Espacio]
GO

/*Sala1 0    2021-03-07 11:17:14.5542431 9999-12-31 23:59:59.9999999
Sala2  0    2021-03-07 11:12:07.7256059 9999-12-31 23:59:59.9999999
Sala3  1    2021-03-07 11:19:59.1486039 9999-12-31 23:59:59.9999999
Sala4  1    2021-03-07 11:12:07.7256059 9999-12-31 23:59:59.9999999*/
```

```
SELECT * FROM [trial].[reserva_sala_pendiente_desinfectar]
GO

/*ID_Espacio      Reserva_Sala      SysStartTime      SysEndTime
Sala1    1    2021-03-07 11:12:07.7256059 2021-03-07 11:17:14.5542431
Sala3    0    2021-03-07 11:12:07.7256059 2021-03-07 11:19:59.1486039
Sala5    1    2021-03-07 11:12:07.7256059 2021-03-07 11:21:25.6961863
Sala5    0    2021-03-07 11:21:25.6961863 2021-03-07 11:25:00.8988244*/
```

Vuelvo a agregarla con una persona que la está ocupando.

```
INSERT INTO [trial].[BBM_Espacio] (ID_Espacio,Reserva_Sala)
VALUES ('Sala5',1)
GO

--(1 row affected)
```

```
SELECT * FROM [trial].[BBM_Espacio]
GO

/*ID_Espacio    Reserva_Sala    SysStartTime    SysEndTime
Sala1     0    2021-03-07 11:17:14.5542431 9999-12-31 23:59:59.9999999
Sala2     0    2021-03-07 11:12:07.7256059 9999-12-31 23:59:59.9999999
Sala3     1    2021-03-07 11:19:59.1486039 9999-12-31 23:59:59.9999999
Sala4     1    2021-03-07 11:12:07.7256059 9999-12-31 23:59:59.9999999
Sala5     1    2021-03-07 11:41:04.1666536 9999-12-31 23:59:59.9999999*/
```

```
SELECT * FROM [trial].[reserva_sala_pendiente_desinfectar]
GO

/*ID_Espacio    Reserva_Sala    SysStartTime    SysEndTime
Sala1     1    2021-03-07 11:12:07.7256059 2021-03-07 11:17:14.5542431
Sala3     0    2021-03-07 11:12:07.7256059 2021-03-07 11:19:59.1486039
Sala5     1    2021-03-07 11:12:07.7256059 2021-03-07 11:21:25.6961863
Sala5     0    2021-03-07 11:21:25.6961863 2021-03-07 11:25:00.8988244*/
```

Ahora pruebo a usar el comando **WAITFOR DELAY**, lo que hace es atrasar la actualización de lo que le marco de **hh:mm:ss**. Tras esa espera, me actualiza todo con los usos históricos y el actual. En este caso, puede ser bueno para los intermedios entre que se marcha un paciente y entre el otro, lo que provocaría a nivel aplicativo que los datos fuesen más reales.

```
UPDATE [trial].[BBM_Espacio]
SET Reserva_Sala = 0
WHERE ID_Espacio = 'Sala5'
WAITFOR DELAY '00:02:00' --> Tiene un retraso en la actualización de
unos 2 minutos
GO

--(1 row affected) --> Tardó dos minutos en actualizarlo

SELECT * FROM [trial].[BBM_Espacio]
GO

/*ID_Espacio    Reserva_Sala    SysStartTime    SysEndTime
Sala1     0    2021-03-07 11:17:14.5542431 9999-12-31 23:59:59.9999999
Sala2     0    2021-03-07 11:12:07.7256059 9999-12-31 23:59:59.9999999
Sala3     1    2021-03-07 11:19:59.1486039 9999-12-31 23:59:59.9999999
Sala4     1    2021-03-07 11:12:07.7256059 9999-12-31 23:59:59.9999999
Sala5     0    2021-03-07 11:44:42.9952808 9999-12-31 23:59:59.9999999*/
```

```
SELECT * FROM [trial].[reserva_sala_pendiente_desinfectar]
GO
```

```
/*ID_Espacio    Reserva_Sala    SysStartTime    SysEndTime
Sala1     1    2021-03-07 11:12:07.7256059 2021-03-07 11:17:14.5542431
Sala3     0    2021-03-07 11:12:07.7256059 2021-03-07 11:19:59.1486039
Sala5     1    2021-03-07 11:12:07.7256059 2021-03-07 11:21:25.6961863
Sala5     0    2021-03-07 11:21:25.6961863 2021-03-07 11:25:00.8988244
Sala5     1    2021-03-07 11:41:04.1666536 2021-03-07 11:44:42.9952808*/
```

En SQL Server 2016, tenemos una nueva cláusula FOR SYSTEM_TIME en la instrucción **SELECT ... FROM**, que tiene cinco nuevas subcláusulas para consultar datos de tabla temporal:

- **AS OF** <datetime>
- **ALL**
- **FROM** <start_datetime> TO <end_datetime>
- **BETWEEN** <start_datetime> AND <end_datetime>
- **CONTAINED IN** (start_datetime, end_datetime)

La subcláusula **AS OF** devuelve filas de la tabla temporal y de historial que son válidas hasta el momento que especifique.

```
SELECT *
FROM [trial].[BBM_Espacio]
FOR SYSTEM_TIME AS OF '2021-03-07 12:19:09.7172414'
GO
/*ID_Espacio    Reserva_Sala    SysStartTime    SysEndTime
Sala1     0    2021-03-07 12:19:09.6235230 9999-12-31 23:59:59.9999999
Sala2     0    2021-03-07 12:19:09.5925306 9999-12-31 23:59:59.9999999
Sala3     1    2021-03-07 12:19:09.7172414 9999-12-31 23:59:59.9999999
Sala4     1    2021-03-07 12:19:09.5925306 9999-12-31 23:59:59.9999999
Sala5     1    2021-03-07 12:19:09.5925306 2021-03-07 12:19:10.0141820*/
```

La subcláusula **FROM-TO** devuelve resultados al combinar la tabla temporal y la de historial, pero no vemos la asignación actual de Eva porque esta cláusula excluye el límite superior del tiempo de finalización. Le brinda la instantánea completa de los valores actuales hasta el momento especificado.

```
SELECT *
FROM [trial].[BBM_Espacio]
FOR SYSTEM_TIME FROM '2021-03-07 12:19:09.6235230' TO '2021-03-07
12:19:09.7172414'
GO

/*ID_Espacio    Reserva_Sala    SysStartTime    SysEndTime
Sala1     0    2021-03-07 12:19:09.6235230 9999-12-31 23:59:59.9999999
Sala2     0    2021-03-07 12:19:09.5925306 9999-12-31 23:59:59.9999999
Sala4     1    2021-03-07 12:19:09.5925306 9999-12-31 23:59:59.9999999
Sala3     0    2021-03-07 12:19:09.5925306 2021-03-07 12:19:09.7172414
Sala5     1    2021-03-07 12:19:09.5925306 2021-03-07 12:19:10.0141820*/
```

Por otro lado, **ALL** le ofrece todo, desde la tabla actual y la historia.

```
SELECT *
FROM [trial].[BBM_Espacio]
FOR SYSTEM_TIME ALL
GO

/*ID_Espacio    Reserva_Sala    SysStartTime    SysEndTime
Sala1     0    2021-03-07 12:19:09.6235230 9999-12-31 23:59:59.9999999
Sala2     0    2021-03-07 12:19:09.5925306 9999-12-31 23:59:59.9999999
Sala3     1    2021-03-07 12:19:09.7172414 9999-12-31 23:59:59.9999999
Sala4     1    2021-03-07 12:19:09.5925306 9999-12-31 23:59:59.9999999
Sala5     0    2021-03-07 12:19:10.5921902 9999-12-31 23:59:59.9999999
Sala1     1    2021-03-07 12:19:09.5925306 2021-03-07 12:19:09.6235230
Sala3     0    2021-03-07 12:19:09.5925306 2021-03-07 12:19:09.7172414
Sala5     1    2021-03-07 12:19:09.5925306 2021-03-07 12:19:10.0141820
Sala5     0    2021-03-07 12:19:10.0141820 2021-03-07 12:19:10.4047081
Sala5     1    2021-03-07 12:19:10.4985184 2021-03-07 12:19:10.5921902*/
```

La subcláusula **BETWEEN-AND** devuelve resultados combinados de las tablas temporal e histórica y podemos ver la asignación actual de Eva. Esto se debe a que esta subcláusula incluye el límite superior del tiempo de finalización.

```
SELECT *
FROM [trial].[BBM_Espacio]
FOR SYSTEM_TIME BETWEEN '2021-03-07 12:19:09.6235230' AND '2021-03-07
12:19:09.7172414'
GO

/*ID_Espacio    Reserva_Sala    SysStartTime    SysEndTime
Sala1     0    2021-03-07 12:19:09.6235230 9999-12-31 23:59:59.9999999
Sala2     0    2021-03-07 12:19:09.5925306 9999-12-31 23:59:59.9999999
Sala3     1    2021-03-07 12:19:09.7172414 9999-12-31 23:59:59.9999999
Sala4     1    2021-03-07 12:19:09.5925306 9999-12-31 23:59:59.9999999
Sala3     0    2021-03-07 12:19:09.5925306 2021-03-07 12:19:09.7172414
Sala5     1    2021-03-07 12:19:09.5925306 2021-03-07 12:19:10.0141820*/
```

Finalmente, la subcláusula **CONTAINED IN** proporciona todas las filas de la tabla histórica incluidas en el rango especificado (solo devuelve datos de la tabla de historial).

```
SELECT *
FROM [trial].[BBM_Espacio]
FOR SYSTEM_TIME CONTAINED IN ('2021-03-07 12:19:09.5925306', '2021-03-07
12:19:10.5921902')
GO

/*ID_Espacio    Reserva_Sala    SysStartTime    SysEndTime
Sala1     1    2021-03-07 12:19:09.5925306 2021-03-07 12:19:09.6235230
Sala3     0    2021-03-07 12:19:09.5925306 2021-03-07 12:19:09.7172414
Sala5     1    2021-03-07 12:19:09.5925306 2021-03-07 12:19:10.0141820
Sala5     0    2021-03-07 12:19:10.0141820 2021-03-07 12:19:10.4047081
Sala5     1    2021-03-07 12:19:10.4985184 2021-03-07 12:19:10.5921902*/
```

Para borrar la tabla, lo que hay que hacer es “apagar” el SYSTEM_VERSIONING, borrar la tabla temporal, y posteriormente la “embebida”.

```
--Para borrar la tabla

/***** Object: Table [trial].[BBM_Espacio]      Script Date: 07/03/2021
12:04:23 *****/
ALTER TABLE [trial].[BBM_Espacio] SET ( SYSTEM_VERSIONING = OFF )
GO

/***** Object: Table [trial].[BBM_Espacio]      Script Date: 07/03/2021
12:04:23 *****/
DROP TABLE [trial].[BBM_Espacio]
GO

/***** Object: Table [trial].[reserva_sala_pendiente_desinfectar]
Script Date: 07/03/2021 12:04:23 *****/
DROP TABLE [trial].[reserva_sala_pendiente_desinfectar]
GO
```

TRIGGERS

Los triggers o disparadores son objetos de la base de datos que ejecutan **acciones** cuando se producen ciertos **eventos** (tanto DML como DDL) (inserciones, modificaciones, borrados, creación de tablas, etc).

Podríamos distinguir tres tipos de niveles:

- A nivel Server
- A nivel Database
- A nivel Table/View

Los triggers como en las restricciones se pueden tener habilitados o deshabilitados (enable/disable)

La forma de los triggers es de la siguiente manera:

```
/*FORMA*/
-- CREATE [OR ALTER] TRIGGER [schema_name .] trigger_name --> corchetes
implica opcionalidad por ello si pongo si no pongo schema pone dbo
-- ON {table | view } --> brackets implican opción. O una u otra
-- [WITH <dml_trigger_option> [,....n]]
--{ FOR | AFTER | INSTEAD OF }
--{ [ INSERT ] [ , ] [ UPDATE ] [ , ] [ DELETE ] }
--[ WITH APPEND ]
--[ NOT FOR REPLICATION ]
--AS { sql_statement [ ; ] [ ,...n ] | EXTERNAL NAME <method specifier [ ;
] > }

--<dml_trigger_option> ::=
-- [ ENCRYPTION ]
--[ EXECUTE AS Clause ]

--<methodSpecifier> ::=
-- assembly_name.class_name.method_name
```

Empiezo por el **trigger de servidor**. Las palabras reservadas **FOR** e **INSTEAD** no se aplican en servidores.

Primero controlo la existencia del *trigger* y lo creo con el comando **CREATE OR ALTER**. Indico que se hace en todos los servidores y para la sentencia **CREATE_LOGIN**, y que me imprima una frase acompañada de un **ROLLBACK** que va a provocar que la sentencia no se ejecute (es el antónimo de **EXECUTE**).

```
USE master
GO

DROP TRIGGER IF EXISTS trg_BBM_FORBIDLOGINS

CREATE OR ALTER TRIGGER trg_BBM_FORBIDLOGINS
ON ALL SERVER --SERVER LEVEL
FOR CREATE_LOGIN -- Sentencia a controlar (puede hacer más de una)
AS --> CUERPO DEL TRIGGER
```

```

PRINT 'Usted no puede crear un LOGIN. Contacte con su manager para
tal fin'
ROLLBACK TRAN --> Provoca que la sentencia no se provoque. Es el
antónimo de EXECUTE
GO

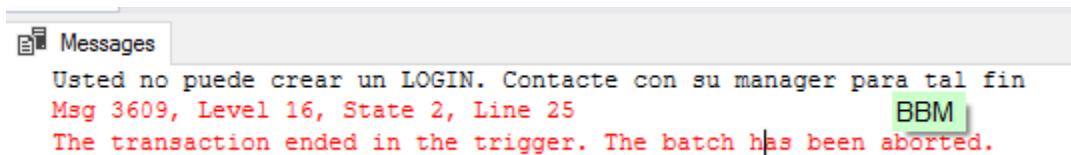
```

Ahora intento hacer la creación de un *login* para ver el efecto. Vemos el mensaje que nos provoca:

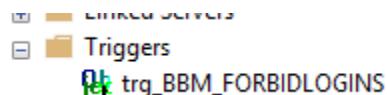
```

CREATE LOGIN USERFORBIDTEST WITH PASSWORD = 'Abcd1234.'
GO

```



A efectos de **Object Explorer** me lo crea en la carpeta **Server Objects > Triggers**. De ahí podría sacarlo con **CREATE** o **New Query Editor**.



Para finalizar lo Deshabilito, lo habilito y lo borro.

```

-- En vez de print podríamos hacer lo mismo con un RAISERROR

DISABLE trigger ALL ON ALL SERVER;
GO

ENABLE Trigger ALL ON ALL SERVER;
GO

-- El trigger está activado. Si queremos borrarlo: DROP
DROP TRIGGER trg_BBM_FORBIDLOGINS
GO

```

NOTA: En caso de que de fallo al borrarlo, lo hago con la GUI

Para esta prueba voy a usar una de las tablas con el **schema trial** que tengo en la base de datos original para hacer pruebas también. La prueba es para ver los triggers a nivel **Base de datos**.

```

USE BBM_ASPACE
GO

SELECT * FROM trial.BBM_Sesion

```

Sesion_ID	Fecha_y_Hora_de_Inicio	Fecha_y_Hora_de_fin	BBM_Horario_ID_Horario	BBM_Espacio_ID_Espacio	BBM_Equipamiento_ID_Equipamiento	BBM_Epleado_BBMM_Epleado_ID	BBM_Actividades_BBMM_Actividades_ID
1	2021-01-24 10:00:00.000	2021-01-24 12:00:00.000			ELECTRO	0	0
2	2021-01-25 10:00:00.000	2021-01-25 11:00:00.000			INDIBA	0	0

Controlo la existencia del trigger. Dejo dos maneras de hacerlo: la *deprecated* y la nueva.

```
--Creamos el trigger (el FOR AFTER, no es necesario, llega con poner FOR)
IF OBJECT_ID('trg_BBM_NODROP', 'TR') IS NOT NULL
    DROP TRIGGER trg_BBM_NODROP
GO

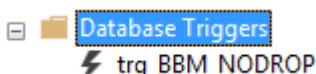
DROP TRIGGER IF EXISTS trg_BBM_NODROP
GO
```

Creo el trigger sobre la base de datos para las sentencias DROP y ALTER para que me de un error de que no se pueden borrar o modificar las tablas, y un Rollback para que no se produzcan.

```
CREATE OR ALTER TRIGGER trg_BBM_NODROP
ON DATABASE
FOR DROP_TABLE, ALTER_TABLE

AS
    RAISERROR ('No se puede borrar o modificar tablas por Usuarios o
Empleados Consulte al departamento de IT',10,1)
    ROLLBACK TRANSACTION
GO
```

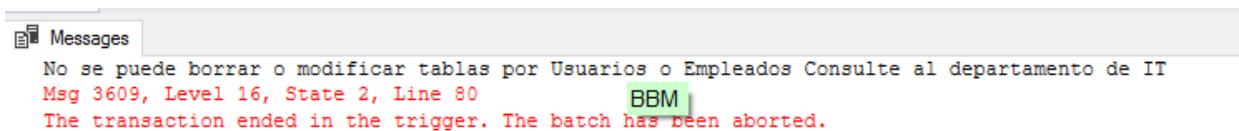
Vemos como se crea en el Object Explorer, en la Base de datos>Programmability>Database Triggers



Intento probar a borrar la tabla y vemos que no me deja.

```
--Probamos a borrar la tabla BBM_Comidas

DROP TABLE trial.BBM_Comidas;
GO
```



Ahora deshabilito el trigger y la borro.

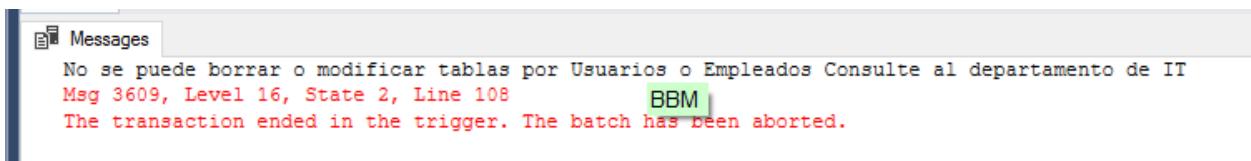
```
DISABLE TRIGGER trg_BBM_NODROP
ON DATABASE;
GO
```

```
DROP TABLE trial.BBM_Comidas;
GO
```

Habilito el trigger y trato de borrar una *constraint* de la tabla BBM_Expediente para probar también el error en los ALTER.

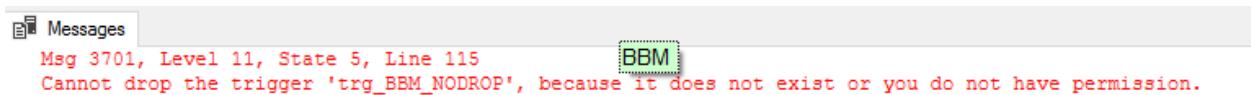
```
ENABLE TRIGGER trg_BBM_NODROP ON DATABASE;
GO

ALTER TABLE [trial].[BBM_Expediente] DROP CONSTRAINT
[BBM_Session_Session_ID_FK]
GO
```

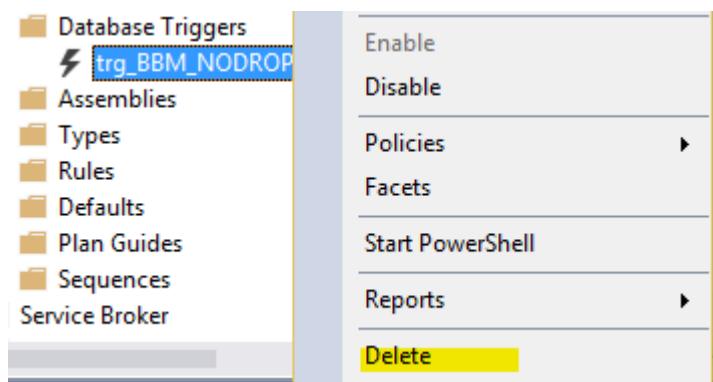


Trato de borrar el trigger.

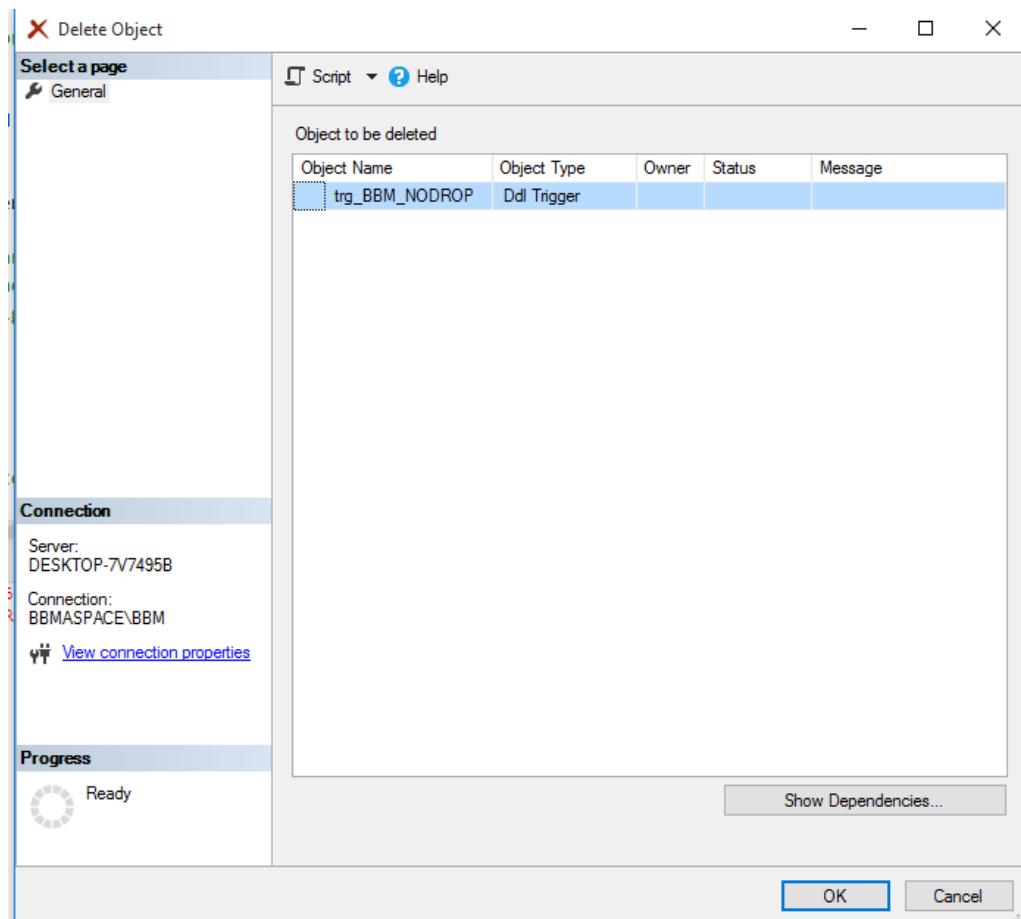
```
DROP TRIGGER trg_BBM_NODROP
GO
```



Como no me deja, lo hago a través de la GUI. Esto se hace con botón derecho sobre el trigger seleccionando Delete.



Sale una ventana de Delete Object, acepto.



Ahora me queda el trigger a nivel Tabla/Vista. Un detalle es que no se controlan triggers de varias tablas en una misma inserción.

Para esto usaré varios ejemplos. Como ya expliqué un poco la mecánica de los triggers, me voy a limitar a poner el resto del script por partes dando una pequeña introducción de cada una de ellas. Para este en concreto voy a meter un trigger que lo que haga es controlar las operaciones de actualización. Primero ejecuta la operación de actualización, da el RAISERROR y a mayores lo ejecuta.

```
use BBM_ASPACE
Go

DROP TABLE IF EXISTS trial.BBM_Comidas
Go

SELECT *
    into trial.BBM_Comidas
    from dbo.BBM_Cómidas
Go

SELECT * FROM trial.BBM_Comidas

--Controlo el trigger
```

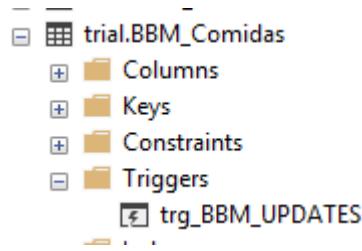
```
DROP TRIGGER IF EXISTS trg_BBM_UPDATES
GO
-- Creamos un trigger que nos ejecute un raiserror y un procedimiento
almacenado
-- Después de una inserción o un update en la tabla trial.BBM_Comidas
CREATE OR ALTER TRIGGER trg_BBM_UPDATES
ON trial.BBM_Comidas
AFTER INSERT, UPDATE --SI ponemos FOR es un 'After' --> hacen la misma
función
AS
    RAISERROR (50009,16,10)
    EXEC sp_helpdb BBM_ASPACE
GO
/*EL TRIGER CONTROLA OPERACIONES DE ACTUALIZACIÓN, LO QUE HACE ES
EJECUTAR UNA OPERACIÓN DE ACTUALIZCIÓN, TE DA EL RAISERROR Y A MAYORES
EXJECUTA*/
--Comprobamos
```

```
Select * from trial.BBM_Comidas;
GO
```

ID_Menu	Plato_1	Plato_2	Plato_3	BBM_Otros_Departamentos_OD_ID

Comprobamos el trigger

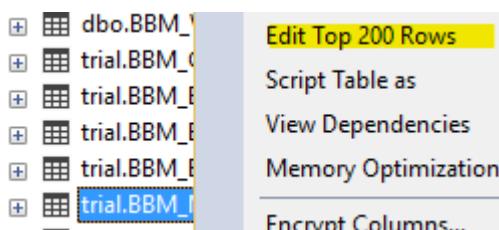
```
--Lo probamos
UPDATE trial.BBM_Comidas
    SET ID_Menu = 'Desayuno'
    Where Plato_1= 'Manzana';
GO
--Msg 18054, Level 16, State 1, Procedure
trg_BBM_UPDATES, Line 5 [Batch Start Line 211]
--Error 50009, severity 16, state 10 was raised, but no message with that
error number was found in sys.messages. If error is larger than 50000,
make sure the user-defined message is added using sp_addmessage.
```



```
Select * from trial.BBM_Comidas;
GO
ID_Menu | Plato_1 | Plato_2 | Plato_3 | BBM_Otros_Departamentos_OD_ID |
DISABLE TRIGGER trg_BBM_UPDATES ON trial.BBM_Comidas
GO
ENABLE TRIGGER trg_BBM_UPDATES ON trial.BBM_Comidas
GO
```

```
DROP TRIGGER trg_BBM_UPDATES
GO
```

En el siguiente *trigger*, voy a hacer la prueba de borrar valores de una tabla. Para ello, le inserto por GUI valores a la tabla trial.BBM_Comidas. Para ello, voy a la tabla y hago clic con el botón derecho sobre ella y elijo Edit Top 200 Rows. Ahí inserto valores:



	ID_Medicacion	Nombre_Medi...	Codigo_EAN	Fabricante	Posoloquia	Descripcion	Observaciones
	NEUNUFOR	NEURO-NUTRI...	0	superSMART	Tomar de 1 a 3 ...		Contiene deriva...
	FREN	FRENADOL	1	Bayern	Tomar cada 24		
	TERMAL	TERMALGIN	2	Bayern	Tomar cada 6 h...	NULL	NULL

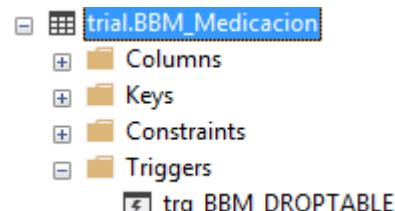
Dejo el resto del script aquí

```
/*PRUEBA DE TRIGGER CON BORRADO DE TABLA*/
```

```

CREATE OR ALTER TRIGGER trg_BBM_DROPTABLE
ON trial.BBM_Medicacion
FOR DELETE, UPDATE
AS
    RAISERROR ('%d row modified in
trial.BBM_Medicacion', 16,1,@@rowcount) --> es
una especie de formato %d
GO

```



```

SELECT*
FROM trial.BBM_Medicacion
WhereCodigo_EAN = '0'
GO

```

	ID_Medicacion	Nombre_Medicamento	Codigo_EAN	Fabricante	Posología	Descripción	Observaciones
1	NEUNUFOR	NEURO-NUTRITION FORMULA	0	superSMART	Tomar de 1 a 3 cápsulas vegetales al día con las...		Contiene derivados de la soja

--Try out

```

DELETE trial.BBM_Medicacion
Where Codigo_EAN= '0';
GO

```

Messages

```

Msg 50000, Level 16, State 1, Procedure trg_BBM_DROPTABLE, Line 5 [Batch Start Line 212]
1 row modified in trial.BBM_Medicacion BBM
(1 row affected)

```

```

--Msg 50000, Level 16, State 1, Procedure trg?borra, Line 5 `Batch Start
Line 246+
--1 filas modificadas en la tabla Autores

```

```

SELECT*
FROM trial.BBM_Medicacion
Where Codigo_EAN = '0'
GO

```

	ID_Medicacion	Nombre_Medicamento	Codigo_EAN	Fabricante	Posología	Descripción	Observaciones

--cerramos el círculo deshabilitando habilitando y borrando

```

DISABLE TRIGGER trg_BBM_DROPTABLE ON trial.BBM_Medicacion
GO

```

```

ENABLE TRIGGER trg_BBM_DROPTABLE ON trial.BBM_Medicacion
GO

```

```

DROP TRIGGER trg_BBM_DROPTABLE
GO

```

Ahora probamos sobre una vista usando instead of:

```
USE BBM_ASPACE
```

```
Go
```

-- Creamos la vista

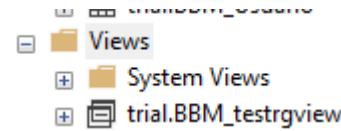
```
CREATE OR ALTER VIEW trial.BBM_testrgview
```

```
AS
```

```
SELECT *
```

```
FROM trial.BBM_Usuario
```

```
GO
```



```
-- Creamos un trigger para la vista (de tipo
INSTEAD OF)

CREATE OR ALTER TRIGGER trg_BBM_DROPtestrgview
ON trial.BBM_testrgview
INSTEAD OF DELETE
AS
PRINT 'Esto es un test de prueba sobre borrados de vista'
GO
SELECT * FROM trial.BBM_testrgview
GO
```

The screenshot shows the SQL query above being run. The results pane displays a table with columns: Usuario_ID, Nombre_1, Nombre_2, Apellido_1, Apellido_2, DNI, and Otros_Detalles. The data is as follows:

	Usuario_ID	Nombre_1	Nombre_2	Apellido_1	Apellido_2	DNI	Otros_Detalles
1	USERGPP012	PACO		GONZALEZ	PEREZ	11111012S	
2	USERVNA013	ANA	MARIA	VAZQUEZ	NUNEZ	11111013N	
3	USERVRI013	ISABEL		VILLAR	RODRIGUEZ	11111013D	Violento

```
DELETE FROM trial.BBM_testrgview;
GO
```

Messages

Esto es un test de prueba sobre borrados de vista
(3 rows affected)

```
select * from trial.BBM_testrgview;
GO
```

The screenshot shows the results of the 'select *' query after the trigger was created. The results pane displays the same table with the same data as the first screenshot.

	Usuario_ID	Nombre_1	Nombre_2	Apellido_1	Apellido_2	DNI	Otros_Detalles
1	USERGPP012	PACO		GONZALEZ	PEREZ	11111012S	
2	USERVNA013	ANA	MARIA	VAZQUEZ	NUNEZ	11111013N	
3	USERVRI013	ISABEL		VILLAR	RODRIGUEZ	11111013D	Violento

```
DROP TRIGGER trg_BorrarVista
GO
```

Ahora hago una segunda prueba de triggers sobre tabla usando After:

```
-Controlo el trigger
IF OBJECT_ID ('Trg_DarAutor', 'TR') IS NOT
NULL
    DROP TRIGGER Trg_BBM_AFTERINSERTUPDATE;
GO

DROP TRIGGER IF EXISTS
trg_BBM_AFTERINSERTUPDAT
GO

-- Creamos un trigger que nos ejecute un raiserror y un procedimiento
almacenado
-- Después de una inserción o un update en la tabla autores
CREATE OR ALTER TRIGGER trg_BBM_AFTERINSERTUPDAT
ON trial.BBM_Expediente
```



```
AFTER INSERT, UPDATE --SI ponemos FOR es un 'After' --> hacen la misma
función
AS
    RAISERROR (50009,16,10)
    EXEC sp_helpdb BBM_ASPACE
GO
/*EL TRIGER CONTROLA OPERACIONES DE ACTUALIZACIÓN, LO QUE HACE ES
EJECUTAR UNA OPERACIÓN DE ACTUALIZCIÓN, TE DA EL RAISERROR Y A MAYORES
EXJECUTA*/
```

--Comprobamos

```
Select * from trial.BBM_Expediente;
GO
```

	ID_Expediente	BBM_Enfermedad_Diagnosticada_Nombre_Enfermedad	BBM_Usuario_Usuario_ID	BBM_Sesion_Sesion_ID	BBM_Tratamientos_BBM_Tratamientos_ID	Especialista_ID
1	GPPTUT012	APRAXIA IDEOMOTORA	USERGPP012	LU25012021	PSIC	NULL
2	TESTEO001	APRAXIA IDEOMOTORA	USERGPP012	DO24012021	FISI	FISIO

--Lo probamos

```
UPDATE trial.BBM_Expediente
SET ID_Expediente = 'INSRTT3ST'
Where BBM_Enfermedad_Diagnosticada_Nombre_Enfermedad = 'TESTEO';
GO
/*Msg 18054, Level 16, State 1, Procedure trg_BBM_AFTERINSERTUPDAT, Line
5 [Batch Start Line 311]
Error 50009, severity 16, state 10 was raised, but no message with that
error number was found in sys.messages.
If error is larger than 50000, make sure the user-defined message is
added using sp_addmessage.
(0 rows affected)
*/
```

	ID_Expediente	BBM_Enfermedad_Diagnosticada_Nombre_Enfermedad	BBM_Usuario_Usuario_ID	BBM_Sesion_Sesion_ID	BBM_Tratamientos_BBM_Tratamientos_ID	Especialista_ID
1	GPPTUT012	APRAXIA IDEOMOTORA	USERGPP012	LU25012021	PSIC	NULL
2	TESTEO001	APRAXIA IDEOMOTORA	USERGPP012	DO24012021	FISI	FISIO

```
Select * from trial.BBM_Expediente;
GO
```

```
DISABLE TRIGGER trg_BBM_AFTERINSERTUPDAT ON trial.BBM_Expediente
GO
```

```
ENABLE TRIGGER trg_BBM_AFTERINSERTUPDAT ON trial.BBM_Expediente
GO
```

```
DROP TRIGGER trg_BBM_AFTERINSERTUPDAT
GO
```

Esta vez la prueba es con un *trigger* que afecte a las actualizaciones y a los borrados con una sentencia **FOR**:

```

CREATE OR ALTER TRIGGER trg_BBM_DELETEUPDATE
ON trial.BBM_Tratamientos
FOR DELETE, UPDATE
AS
    RAISERROR ('%d filas modificadas en la
    tabla Autores', 16,1,@@rowcount) --> es una
    especie de formato %d
GO

SELECT*
FROM trial.BBM_Tratamientos
Where Tratamiento_ID = '141'
GO

--Try out
DELETE trial.BBM_Tratamientos
Where Tratamiento_ID= '141';
GO

Msg 50000, Level 16, State 1, Procedure trg_BBM_DELETEUPDATE, Line 5 [Batch Start Line 353]
1 filas modificadas en la tabla Autores
(1 row affected)

SELECT*
FROM trial.BBM_Tratamientos
Where Tratamiento_ID = '141'
GO

--cerramos el circulo deshabilitando habilitando y borrando

DISABLE TRIGGER trg_BBM_DELETEUPDATE ON trial.BBM_Tratamientos
GO

ENABLE TRIGGER trg_BBM_DELETEUPDATE ON trial.BBM_Tratamientos
GO

DROP TRIGGER trg_BBM_DELETEUPDATE
GO

```

Como se ha visto en los ejemplos que he puesto arriba, Los *triggers* pueden definirse para las operaciones **INSERT**, **DELETE** o **UPDATE**, y pueden dispararse antes o después de la operación. Finalmente, el nivel de los disparadores puede ser la fila o registro o la orden.

El modificador **BEFORE** o **AFTER** indica que el *trigger* se ejecutará antes o después de ejecutarse la sentencia SQL definida por **DELETE**, **INSERT** o **UPDATE**. Si incluimos el modificador **OF** el *trigger* solo se ejecutará cuando la sentencia SQL afecte a los campos incluidos en la lista.

El alcance de los disparadores puede ser la fila o de orden. El modificador **FOR EACH ROW** indica que el *trigger* se disparará cada vez que se realizan operaciones sobre cada

fila de la tabla. Si se acompaña del modificador **WHEN**, se establece una restricción; el trigger solo actuará, sobre las filas que satisfagan la restricción.

Categoría	Valores	Comentarios
Orden	INSERT, DELETE, UPDATE	Define que tipo de operación DML provoca la activación del trigger
Temporalización	BEFORE / AFTER	Define si el disparador se activa antes o después de que se ejecute la operación DML
Nivel	Fila Orden /	Los disparadores con nivel de fila se activan una vez por cada fila afectada por la orden que provocó el disparo. Los Triggers con nivel de orden se activan sólo una vez, antes o después de la orden. Los disparadores con nivel de fila se identifican por la cláusula FOR EACH ROW en la definición del disparador.

Una misma tabla puede tener varios *triggers* asociados. En tal caso es necesario conocer el orden en el que se van a ejecutar. Los disparadores se activan al ejecutarse la sentencia SQL.

- Si existe, se ejecuta el disparador de tipo **BEFORE** (disparador previo) con nivel de orden.
- Para cada fila a la que afecte la orden:
- Se ejecuta si existe, el disparador de tipo **BEFORE** con nivel de fila.
- Se ejecuta la propia orden.
- Se ejecuta si existe, el disparador de tipo **AFTER** (disparador posterior) con nivel de fila.
- Se ejecuta, si existe, el disparador de tipo **AFTER** con nivel de orden.

En las instrucciones de desencadenadores DML se usan dos tablas especiales: la tabla **inserted** y la tabla **deleted**. SQL Server crea y administra automáticamente ambas tablas. Puede utilizar estas tablas temporales residentes en memoria para probar los efectos de determinadas modificaciones de datos y para establecer condiciones para las acciones de los desencadenadores DML. No puede modificar directamente los datos de estas tablas ni realizar en ellas operaciones de lenguaje de definición de datos (DDL), como **CREATE INDEX**.

La tabla **deleted** almacena copias de las filas afectadas por las instrucciones **DELETE** y **UPDATE**. Durante la ejecución de una instrucción **DELETE** o **UPDATE**, las filas se eliminan de la tabla del desencadenador y se transfieren a la tabla **deleted**. La tabla **deleted** y la tabla del desencadenador no suelen tener filas en común.

La tabla **inserted** almacena copias de las filas afectadas durante las instrucciones **INSERT** y **UPDATE**. Durante una transacción de inserción o actualización, se agregan nuevas filas a la tabla **inserted** y a la tabla del desencadenador. Las filas de la tabla **inserted** son copias de las nuevas filas de la tabla del desencadenador.

Después de esta breve explicación, introduzco las dos sentencias que me faltan: **AFTER** e **INSTEAD OF**. **FOR** o **AFTER** especifica que el desencadenador DML solo se activa cuando todas las operaciones especificadas en la instrucción SQL desencadenadora se han iniciado correctamente. Además, todas las acciones referenciales en cascada y las comprobaciones de restricciones también deben ser correctas para que este desencadenador se active.

```
DROP TRIGGER IF EXISTS trg_BBM_AFTERUPDATE
```

```
GO
```

```
CREATE OR ALTER TRIGGER trg_BBM_AFTERUPDATE
```

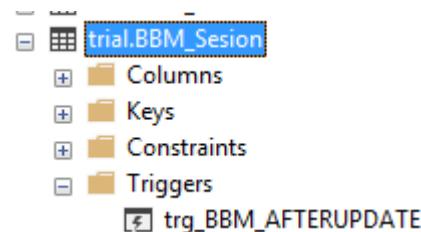
```
ON trial.BBM_Sesion
```

```
AFTER UPDATE
```

```
AS
```

```
PRINT 'Tabla inserted'  
select * from inserted  
PRINT 'Tabla inserted'  
select * from deleted
```

```
Go
```



```
select * from trial.BBM_Sesion
```

```
GO
```

Sesion_ID	Fecha_y_Hora_de_Inicio	Fecha_y_Hora_de_fin	BBM_Horario_ID_Horario	BBM_Espacio_ID_Espacio	BBM_Equipamiento_ID_Equipamiento	BBM_Epleado_BBM_Epleado_ID	BBM_Actividades_BBM_Actividades_ID
1	2021-01-24 10:00:00.000	2021-01-24 12:00:00.000			ELECTRO	0	0
2	2021-01-25 10:00:00.000	2021-01-25 11:00:00.000			INDIBA	0	0

```
UPDATE trial.BBM_Sesion
```

```
SET BBM_Horario_ID_Horario = '10001200'
```

```
WHERE Sesion_ID = 'DO24012021';
```

```
GO
```

Sesion_ID	Fecha_y_Hora_de_Inicio	Fecha_y_Hora_de_fin	BBM_Horario_ID_Horario	BBM_Espacio_ID_Espacio	BBM_Equipamiento_ID_Equipamiento	BBM_Epleado_BBM_Epleado_ID	BBM_Actividades_BBM_Actividades_ID
1	2021-01-24 10:00:00.000	2021-01-24 12:00:00.000	10001200		ELECTRO	0	0

INSERTED

Sesion_ID	Fecha_y_Hora_de_Inicio	Fecha_y_Hora_de_fin	BBM_Horario_ID_Horario	BBM_Espacio_ID_Espacio	BBM_Equipamiento_ID_Equipamiento	BBM_Epleado_BBM_Epleado_ID	BBM_Actividades_BBM_Actividades_ID
1	2021-01-24 10:00:00.000	2021-01-24 12:00:00.000			ELECTRO	0	0

DELETED

```
SELECT * FROM trial.BBM_Sesion
```

```
GO
```

	Sesion_ID	Fecha_y_Hora_de_inicio	Fecha_y_Hora_de_fin	BBM_Horario_ID_Horario	BBM_Espacio_ID_Espacio	BBM_Equipamiento_ID_Equipamiento	BBM_Epleado_BBMM_Epleado_ID	BBM_Actividades_BBMM_Actividades_ID
1	DO24012021	2021-01-24 10:00:00.000	2021-01-24 12:00:00.000	10001200		ELECTRO	0	0
2	LU25012021	2021-01-25 10:00:00.000	2021-01-25 11:00:00.000			INDIBA	0	0

INSTEAD OF especifica que se inicia el desencadenador DML en lugar de la instrucción SQL desencadenadora, por lo que se sobrescriben las acciones de las instrucciones desencadenadoras. No puede especificar **INSTEAD OF** para los desencadenadores DDL o LOGON.

Como máximo, puede definir un desencadenador **INSTEAD OF** por cada instrucción **INSERT**, **UPDATE** o **DELETE** en una tabla o vista. También puede definir otras vistas en las vistas que tengan su propio desencadenador **INSTEAD OF**.

No puede definir desencadenadores **INSTEAD OF** en vistas actualizables que usan **WITH CHECK OPTION**. Al hacerlo se genera un error cuando se agrega un desencadenador **INSTEAD OF** a una vista actualizable para la que se ha especificado **WITH CHECK OPTION**. Puede quitar esta opción mediante **ALTER VIEW** antes de definir el desencadenador **INSTEAD OF**. En este ejemplo, voy a usar el *trigger* sobre una **vista**.

```
USE BBM_ASPACE
GO

-- Creamos la vista
CREATE OR ALTER VIEW vista_TESTTRIGGER
AS
    SELECT *
    FROM trial.BBM_Tratamientos
GO

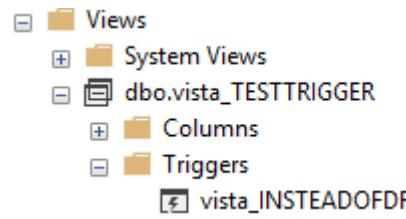
-- Creamos un trigger para la vista (de tipo INSTEAD OF)

CREATE OR ALTER TRIGGER vista_INSTEADOFDROP
ON vista_TESTTRIGGER
    INSTEAD OF DELETE
AS
    PRINT 'La vista no puede ser borrada. Consulte con el departamento
correspondiente'
GO

SELECT * FROM vista_TESTTRIGGER
GO

Results Messages
BBM_Tratamientos_ID Tratamiento_ID BBM_Auxiliar_BBMM_Epleado_BBMM_Epleado_ID BBM_Especialista_BBMM_Epleado_BBMM_Epleado_ID
1 FISI FISI 123 0
2 PSIC 235 0 0
3 TERO 141 0 0

DELETE FROM vista_TESTTRIGGER;
GO
```



Messages
La vista no puede ser borrada. Consulte con el departamento correspondiente
(3 rows affected)

```
select * from vista_TESTTRIGGER;
GO
```

Results Messages

	BBM_Tratamientos_ID	Tratamiento_ID	BBM_Auxiliar_BBM_Empleado_BBM_Empleado_ID	BBM_Especialista_BBM_Empleado_BBM_Empleado_ID
1	FISI	FISI	123	0
2	PSIC		235	0
3	TERO		141	0

```
DROP TRIGGER vista_INSTEADOFDROP
GO
```

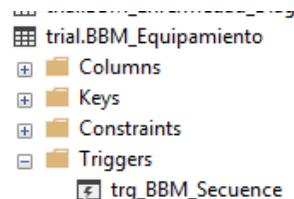
Algunas tablas de la base de datos incluyen una columna de "última modificación", que almacena la fecha y hora en que la fila se actualizó por última vez. Cada vez que se actualiza la fila, la fecha se actualiza para reflejar la fecha y hora de esa actualización. En mi proyecto, esta tabla sería muy adecuada para el Equipamiento, y saber cuándo se coge:

```
USE BBM_ASPACE
GO
DROP TABLE IF EXISTS trial.BBM_Equipamiento
GO
CREATE TABLE trial.BBM_Equipamiento (
    ID_Equipamiento int IDENTITY(1,1) NOT NULL
PRIMARY KEY,
    Nombre_Equipo nvarchar(1000) NOT NULL,
    CreateDate datetime DEFAULT CURRENT_TIMESTAMP,
    ModifiedDate datetime DEFAULT CURRENT_TIMESTAMP
);
GO

CREATE OR ALTER TRIGGER trg_BBM_Secuence
ON trial.BBM_Equipamiento
AFTER UPDATE
AS
    UPDATE trial.BBM_Equipamiento
    SET ModifiedDate = CURRENT_TIMESTAMP
    WHERE ID_Equipamiento IN (SELECT DISTINCT ID_Equipamiento FROM
inserted);
GO

INSERT INTO trial.BBM_Equipamiento(Nombre_Equipo)
VALUES ('Electro');
GO

SELECT * FROM trial.BBM_Equipamiento;
GO
```



```
1   Electro 2021-03-12 22:11:21.113 2021-03-12 22:11:21.113*/  
  
UPDATE trial.BBM_Equipamiento  
SET Nombre_Equipo = 'Electro'  
WHERE ID_Equipamiento = 1;  
GO  
  
SELECT * FROM trial.BBM_Equipamiento;  
GO  
  
/*ID_Equipamiento  Nombre_Equipo  CreateDate  ModifiedDate  
1      Electro 2021-03-12 22:11:21.113 2021-03-12 22:12:46.747*/
```

La explicación es que cuando se ejecuta, actualiza la columna **ModifiedDate** a **CURRENT_TIMESTAMP** (pero solo en la fila que se actualizó). Aquí, La tabla insertada almacena copias de las filas afectadas durante las declaraciones **INSERT** y **UPDATE**. Durante una transacción de inserción o actualización, se agregan nuevas filas tanto a la tabla insertada como a la tabla de activación. Las filas de la tabla insertada son copias de las nuevas filas de la tabla de activación.

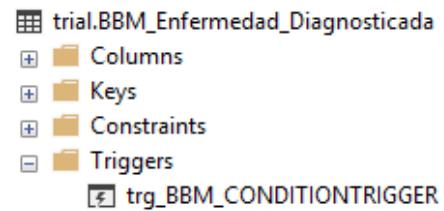
Además de la tabla insertada, SQL Server también crea y mantiene una tabla eliminada. Una transacción de actualización es similar a una operación de eliminación seguida de una operación de inserción; las filas antiguas se copian primero en la tabla eliminada y luego las filas nuevas se copian en la tabla desencadenante y en la tabla insertada.

```
DROP TRIGGER trg_BBM_Secuence  
GO
```

Ahora empiezo a complicar un poco el *script* con un trigger con formato condicional.

```
USE BBM_ASPACE
GO
```

```
SELECT * FROM
trial.BBM_Enfermedad_Diagnosticada
```



```
--Usamos condicional para borrar el trigger
IF OBJECT_ID ('trg_BBM_CONDITIONTRIGGER', 'TR') IS NOT NULL
    DROP TRIGGER trg_BBM_CONDITIONTRIGGER;
GO
```

```
CREATE OR ALTER TRIGGER trg_BBM_CONDITIONTRIGGER
ON trial.BBM_Enfermedad_Diagnosticada
FOR UPDATE
AS
    IF UPDATE (Nombre_Comun_Enfermedad)
        BEGIN
            RAISERROR('No puedes cambiar la tabla', 15,1)
            ROLLBACK TRAN
        END
    ELSE
        PRINT 'Operación Fallida'
GO

SELECT * FROM trial.BBM_Enfermedad_Diagnosticada
GO
```

```
-- No cambia Nombre_Comun_Enfermedad
UPDATE trial.BBM_Enfermedad_Diagnosticada
SET Descripcion= 0
WHERE Nombre_Comun_Enfermedad='Oakland' ;
GO
```

```
SELECT *
FROM trial.BBM_Enfermedad_Diagnosticada
WHERE Nombre_Comun_Enfermedad='APRAXIA'
GO
```

	Nombre_Enfermedad	Nombre_Comun_Enfermedad	Descripcion	Sintomas
1	APRAXIA IDEOMOTORA	APRAXIA	Incapacidad para realizar tareas que requieren r...	no pueden recordar ni hacer la secuencia de movi...
2	PARALISIS CEREBRAL	PA	Lesión en el cerebro que afecta a la movilidad y ...	VARIOS

```
-- NO Cambia Nombre_Comun_Enfermedad

update trial.BBM_Enfermedad_Diagnosticada
set Nombre_Comun_Enfermedad='TEST'
where Nombre_Enfermedad='APRAXIA IDEOMOTORA'
GO

SELECT * FROM trial.BBM_Enfermedad_Diagnosticada
Go
```

Results Messages

	Nombre_Enfermedad	Nombre_Comun_Enfermedad	Descripcion	Sintomas
1	APRAXIA IDEOMOTORA	APRAXIA	Incapacidad para realizar tareas que requieren r...	no pueden recordar ni hacer la secuencia de movi...
2	PARALISIS CEREBRAL	PA	Lesión en el cerebro que afecta a la movilidad y ...	VARIOS

En este ejemplo, voy a intentar que no se pueda actualizar un campo determinado sin la supervisión de un empleado. Lo ideal sería hacerlo sobre una vista, y tener varias vistas de empleados al respecto.

Para ello, empleado el formato condicional de manera que si alguien actualiza el campo *Otros_Detalles*, se imprime un mensaje y se hace **ROLLBACK TRANSACTION**:

```
USE BBM_ASPACE
GO

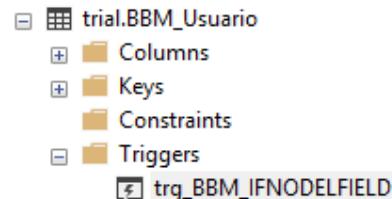
SELECT * FROM trial.BBM_Usuario
GO
```

Results Messages

	Usuario_ID	Nombre_1	Nombre_2	Apellido_1	Apellido_2	DNI	Otros_Detalles
1	USERGPP012	PACO		GONZALEZ	PEREZ	11111012S	
2	USERVNA013	ANA	MARIA	VAZQUEZ	NUNEZ	11111013N	
3	USERVRI013	ISABEL		VILLAR	RODRIGUEZ	11111013D	Violento

```
create OR ALTER trigger trg_BBM_IFNODELFIELD
on trial.BBM_Usuario
after update --after y for se pueden emplear
indistintamente
as
    if UPDATE(Otros_Detalles)
        BEGIN
            PRINT 'Este tipo de
actualizaciones requieren la supervisión de un Empleado Especialista';
            ROLLBACK TRAN;
        END
    GO

--Intentamos actualizar Los Detalles sobre un usuario
UPDATE trial.BBM_Usuario
SET Otros_Detalles='Miedo a la Oscuridad'
WHERE Usuario_ID='USERVNA013';
```



Messages

```
Este tipo de actualizaciones requieren la supervisión de un Empleado Especialista
Msg 3609, Level 16, State 1, Line 20
The transaction ended in the trigger. The batch has been aborted.

--Vemos que no ha actualizado nada
select * from trial.BBM_Usuario WHERE Usuario_ID='USERVNA013';
```

Results

	Usuario_ID	Nombre_1	Nombre_2	Apellido_1	Apellido_2	DNI	Otros_Detalles
1	USERVNA013	ANA	MARIA	VAZQUEZ	NUNEZ	11111013N	

```
DISABLE TRIGGER trg_BBM_IFNODELFIELD ON trial.BBM_Usuario
DROP TRIGGER trg_BBM_IFNODELFIELD
```

En este nuevo ejemplo, meto un **formato condicional** más avanzado, donde **si no se cumple la condición**, entonces (**else**) **realiza otra acción**. En este caso, se va a controlar que no se pueda borrar más de un usuario a la vez por si existe algún tipo de error en los borrados.

```
USE BBM_ASPACE
GO

SELECT * FROM trial.BBM_Usuario ORDER BY Apellido_1
GO
```

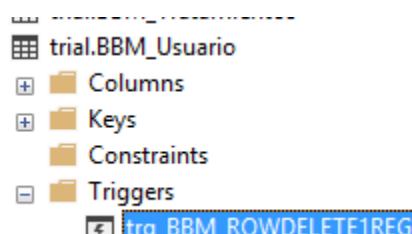
Results

	Usuario_ID	Nombre_1	Nombre_2	Apellido_1	Apellido_2	DNI	Otros_Detalles
1	USERGPP012	PACO		GONZALEZ	PEREZ	11111012S	
2	USERVNA013	ANA	MARIA	VAZQUEZ	NUNEZ	11111013N	
3	USERVRI013	ISABEL		VILLAR	RODRIGUEZ	11111013D	Violento

```
-- Creo el trigger
DROP TRIGGER IF EXISTS trg_BBM_ROWDELETE1REG
GO

CREATE OR ALTER TRIGGER trg_BBM_ROWDELETE1REG
ON trial.BBM_Usuario
FOR DELETE
AS
    IF (@@ROWCOUNT>1) -- Si es más de uno en el recuento
        BEGIN
            RAISERROR('Los registros SOLO se BORRAN de 1 EN 1',15,1)
            ROLLBACK TRAN
        END
    ELSE -- De lo contrario
        PRINT 'Operación Realizada'
GO

DELETE trial.BBM_Usuario
```



GO

Messages

```
Msg 50000, Level 15, State 1, Procedure trg_BBM_ROWDELETE1REG, Line 7 [Batch Start Line 28]
Los registros SOLO se BORRAN de 1 EN 1
Msg 3609, Level 16, State 1, Line 29
The transaction ended in the trigger. The batch has been aborted.
```

```
DELETE trial.BBM_Usuario
WHERE Apellido_1='VILLAR'
GO
```

--Operación Realizada

--(1 row affected)

```
-----
```

```
SELECT * FROM trial.BBM_Usuario ORDER BY Apellido_1
GO
```

Results

Messages

	Usuario_ID	Nombre_1	Nombre_2	Apellido_1	Apellido_2	DNI	Otros_Detalles
1	USERGPP012	PACO		GONZALEZ	PEREZ	11111012S	
2	USERVNA013	ANA	MARIA	VAZQUEZ	NUNEZ	11111013N	

Entre todo el script anterior, cabe destacar un punto pequeño... La palabra **ROWCOUNT**. **ROWCOUNT** devuelve el número de filas afectadas por el último estamento (si ese estamento supera los dos mil millones, hay que usar **ROWCOUNT_BIG**).

En este script, pondré a mayores **TRANCOUNT**, el cual me devuelve la cuenta de las transacciones abiertas en la sesión actual. Se incrementa su valor cuando una transacción se abre, y decrementa cuando se hace "commit" de otra. **ROLLBACK** vuelve el contador de **TRANCOUNT** a cero.

En este script vuelvo a usar el formato condicional para que no se puedan borrar usuarios... ¿Cómo? Poniendo el **ROWCOUNT** a cero y que el **TRANCOUNT** sea siempre **mayor que cero**.

```
USE BBM_ASPACE

SELECT * from trial.BBM_Especialista
go

INSERT INTO trial.BBM_Especialista
    VALUES ('TEST1'),('TEST2'),('TEST3')
GO

--(3 rows affected)
SELECT * from trial.BBM_Especialista
go
```

	Especialista_ID
1	FISIO
2	TEST1
3	TEST2
4	TEST3

```

CREATE OR ALTER TRIGGER trg_BBMROWTRANNODELETE
ON trial.BBM_Especialista
INSTEAD OF DELETE
AS
BEGIN
    DECLARE @Count int;

    SET @Count = @@ROWCOUNT;
    IF @Count = 0
        RETURN;

    BEGIN
        RAISERROR
            ('ESTA TABLA SOLO PUEDE SER ALTERADA POR DBA', -- Message
             10, -- Severity.
             1); -- State.

        -- Rollback any active or uncommittable transactions
        IF @@TRANCOUNT > 0
        BEGIN
            ROLLBACK TRANSACTION;
        END
    END;
END;
GO

-- PRUEBA

DELETE trial.BBM_Especialista
GO

```

Messages

```

ESTA TABLA SOLO PUEDE SER ALTERADA POR DBA
Msg 3609, Level 16, State 1, Line 47
The transaction ended in the trigger. The batch has been aborted.

```

```

DELETE trial.BBM_Especialista
    WHERE Especialista_ID = 'TEST1'-- EXISTE
GO

```

Messages

```

ESTA TABLA SOLO PUEDE SER ALTERADA POR DBA
Msg 3609, Level 16, State 1, Line 57
The transaction ended in the trigger. The batch has been aborted.

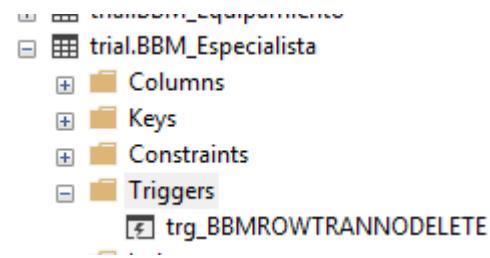
```

```

DELETE trial.BBM_Especialista
    WHERE Especialista_ID = 'TEST700' -- NO EXISTE
GO

--(0 rows affected)

```



Tras comprobar que ambos condicionales se demuestra que se ejecuta, la tabla queda de la siguiente manera:

```
USE BBM_ASPACE
GO
```

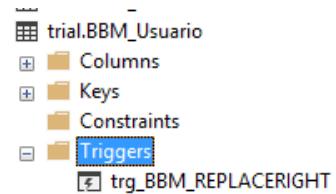
```
SELECT * FROM trial.BBM_Usuario
GO
```

	Usuario_ID	Nombre_1	Nombre_2	Apellido_1	Apellido_2	DNI	Otros_Detalles
1	USERGPP012	PACO		GONZALEZ	PEREZ	11111012S	
2	USERVNA013	ANA	MARIA	VAZQUEZ	NUNEZ	11111013N	

```
-- AHORA CONTROLO LA EXISTENCIA DEL TRIGGER
```

```
IF OBJECT_ID('trg_BBM_REPLACERIGHT','TR') IS NOT NULL
    DROP TABLE trg_BBM_REPLACERIGHT
GO

CREATE OR ALTER TRIGGER trg_BBM_REPLACERIGHT
ON trial.BBM_Usuario -- Nombre de la tabla o vista
INSTEAD OF INSERT -- Recordar que si le pongo after no evita la operacion
a no ser que se le ponga un Rollback
AS
BEGIN --PRINCIPIO DEL CODIGO
    IF EXISTS -- SI EXISTE
        (
            SELECT Nombre_1 -- SIEMPRE VIENE DETRÁS DE UNA SELECT
            FROM Inserted
            WHERE RIGHT(Nombre_1,3) = 'EST' -- SI TOMO LOS TRES PRIMEROS
VALORES POR LA DERECHA
        ) -- SI DEVUELVE FILAS, SE EJECUTA EL INSERT
        INSERT INTO trial.BBM_Usuario
            (Nombre_1, Nombre_2, Apellido_1, Apellido_2, DNI,
Otros_Detalles, Usuario_ID)
            SELECT REPLACE(Nombre_1, 'EST', 'NOMBRETEST'), Nombre_2,
Apellido_1, Apellido_2, DNI, Otros_Detalles, Usuario_ID--> ANTES DE
INSERTAR, REEMPLAZAME AVE POR AVENUE
            FROM Inserted;
    ELSE -- EN CASO DE QUE NO DEVUELVA FILAS, DEVUELVE ESA PARTE --> UN
INSERT SELECT
        INSERT INTO trial.BBM_Usuario
            (Nombre_1, Nombre_2, Apellido_1, Apellido_2, DNI,
Otros_Detalles, Usuario_ID)
            SELECT Nombre_1, Nombre_2, Apellido_1, Apellido_2, DNI,
Otros_Detalles, Usuario_ID --> ¿DE DONDE SACO EL CONTENIDO?
            FROM inserted; --> DE LA TABLA INSERTED
END -- FIN DEL CODIGO
GO
```



```
/*AHORA VAMOS A INSERTAR UN CONTENIDO QUE PROVOQUE EL REEMPLAZO Y OTRO
QUE NO*/
```

```
--COMO SE CUMPLE LA CONDICION IF, ME INSERTA EL NOMBRETEST
```

```
INSERT INTO trial.BBM_Usuario (Nombre_1, Nombre_2, Apellido_1,
Apellido_2, DNI, Otros_Detalles, Usuario_ID)
VALUES
('EST','NOMBRE2','APELIDO1','APELIDO2','123456789','LOREM','USERTEST')
-- Como se cumple la condición, va a hacer el la primera parte del IF
exists
GO
--(1 row affected)

SELECT *
FROM trial.BBM_Usuario
WHERE Nombre_1 = 'NOMBRETEST';
GO
```

	Usuario_ID	Nombre_1	Nombre_2	Apellido_1	Apellido_2	DNI	Otros_Detalles
1	USERTEST	NOMBRETEST	NOMBRE2	APELLIDO1	APELLIDO2	123456789	LOREM

```
INSERT INTO trial.BBM_Usuario (Nombre_1, Nombre_2, Apellido_1,
Apellido_2, DNI, Otros_Detalles, Usuario_ID)
VALUES
('RICHI','NOMBRE2','APELIDO1','APELIDO2','123456789','LOREM','NO
CUMPLE'); -- Como no se cumple la condición, lo que coge es rez de Juan
Florez, entonces hace el Else
GO
```

```
--(1 row affected)
```

```
--(1 row affected)
```

```
SELECT *
FROM trial.BBM_Usuario
WHERE Usuario_ID = 'NO CUMPLE';
GO
```

```
--> Se cumple que lo hace con la tabla Inserted porque no reconoce EST
```

	Usuario_ID	Nombre_1	Nombre_2	Apellido_1	Apellido_2	DNI	Otros_Detalles
1	NO CUMPLE	RICHI	NOMBRE2	APELLIDO1	APELLIDO2	123456789	LOREM

NOTA: Los triggers que faltan no se adaptan al proyecto que desempeño

TABLAS IN MEMORY

En el último *trigger* que he hecho, se ha visto un nuevo concepto no explicado de **tablas in memory**, lo que me hace preguntarme... ¿Qué son?

Las **tablas optimizadas para memoria** son totalmente durables y, al igual que las transacciones en tablas basadas en disco (tradicionales), las transacciones en este tipo de tablas tienen todas las propiedades ACID (atomicidad, coherencia, aislamiento y durabilidad). Las tablas optimizadas para memoria y los procedimientos almacenados compilados de forma nativa admiten solo un subconjunto de características de Transact-SQL.

A partir de SQL Server 2016, y en Azure SQL Database, no existen limitaciones para intercalaciones o páginas de códigos que son específicas de OLTP en memoria.

El almacenamiento principal para las tablas optimizadas para memoria es la memoria principal. Las filas de la tabla se leen y se escriben en la memoria. Una segunda copia de los datos de la tabla se conserva en el disco, pero solo por la durabilidad. Los datos de las tablas optimizadas para memoria solo se leen del disco durante la recuperación de la base de datos (por ejemplo, después de reiniciar el servidor).

Para conseguir incluso mayores mejoras de rendimiento, OLTP en memoria admite tablas durables con la durabilidad diferida de las transacciones. Las transacciones durables diferidas se guardan en el disco en cuanto la transacción se ha confirmado y se devuelve el control al cliente. A cambio de un mayor rendimiento, las transacciones confirmadas que no se hayan guardado en el disco se pierden si el servidor se bloquea o comuta por error.

Además de las tablas optimizadas para memoria durables predeterminadas, también admite tablas optimizadas para memoria no durables, que no se registran y cuyos datos no se guardan en el disco. Esto significa que las transacciones en estas tablas no requieren ninguna E/S de disco, pero los datos no se recuperarán si hay un bloqueo o una comutación por error del servidor.

Las tablas optimizadas para memoria **son siempre con control de versiones**. Esto significa que cada fila de la tabla puede tener varias versiones. Todas las versiones de fila se mantienen en la misma estructura de datos de la tabla. El control de versiones de fila se utiliza para permitir las lecturas y las escrituras simultáneas en la misma fila. Para obtener más información sobre las lecturas y las escrituras simultáneas en la misma fila.

Para crear las tablas, primero crearé una base de datos **TRIALBBM**, que es la que uso siempre para este tipo de pruebas, y que luego implementaría en el proyecto. Como siempre, empiezo controlando su existencia, y luego le asigno los archivos **.mdf** y **.ldf**.

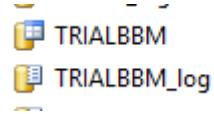
```

USE master
GO

DROP DATABASE IF EXISTS TRIALBBM
GO

CREATE DATABASE [TRIALBBM]
CONTAINMENT = NONE
ON PRIMARY
( NAME = N'TRIALBBM',
FILENAME = N'C:\Program Files\Microsoft SQL
Server\MSSQL14.MSSQLSERVER\MSSQL\DATA\TRIALBBM.mdf' ,
SIZE = 4288KB ,
MAXSIZE = UNLIMITED,
FILEGROWTH = 1024KB )
LOG ON
( NAME = N'TRIALBBM_log',
FILENAME = N'C:\Program Files\Microsoft SQL
Server\MSSQL14.MSSQLSERVER\MSSQL\DATA\TRIALBBM_log.ldf' ,
SIZE = 1072KB ,
MAXSIZE = 2048GB ,
FILEGROWTH = 10%)
GO

```



Ahora modiflico la base de datos y le agrego *FILEGROUPS*. Para ello, necesito más de uno, y además que conetngan el comando **CONTAINS MEMORY_OPTIMIZED_DATA**:

```

USE [TRIALBBM]
GO

ALTER DATABASE [TRIALBBM]
ADD FILEGROUP
TRIALBBM_mod
CONTAINS MEMORY_OPTIMIZED_DATA
GO

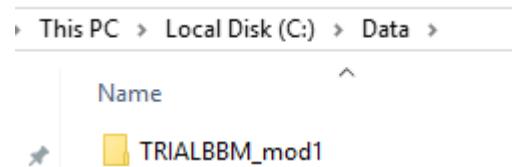
-- You need to add one or more containers to the MEMORY_OPTIMIZED_DATA
filegroup

ALTER DATABASE [TRIALBBM]
ADD FILE (name='TRIALBBM_mod1',
filename='c:\data\TRIALBBM_mod1')
TO FILEGROUP TRIALBBM_mod
GO

-- Look up DB Properties FILEGROUP

```

Database files:				
Logical Name	File Type	Filegroup	Initial Size (MB)	Autogrowth / Maxsize
TRIALBBM	ROWS...	PRIMARY	8	By 1 MB, Unlimited
TRIALBBM_...	FILEST...	TRIALBBM_...	156	Unlimited
TRIALBBM_...	LOG	Not Applicable	70	By 10 percent, Limited to 209...



Para establecer en Transact-SQL los comportamientos del procesamiento de consultas para que sean compatibles con la versión especificada del motor de SQL, se ha de alterar la compatibilidad de la base de datos. En este caso, es la versión de SQL

Server con la que la base de datos se va a hacer compatible lo que marca esta pauta.
Vemos la tabla que lo marca:

Producto	Versión del motor de base de datos	Designación de nivel de compatibilidad predeterminado	Valores de nivel de compatibilidad admitidos
SQL Server 2019 (15.x)	15	150	150, 140, 130, 120, 110, 100
SQL Server 2017 (14.x)	14	140	140, 130, 120, 110, 100
Azure SQL Database	12	150	150, 140, 130, 120, 110, 100
Instancia administrada de Azure SQL Database	12	150	150, 140, 130, 120, 110, 100
SQL Server 2016 (13.x)	13	130	130, 120, 110, 100
SQL Server 2014 (12.x)	12	120	120, 110, 100
SQL Server 2012 (11.x)	11	110	110, 100, 90
SQL Server 2008 R2	10.5	100	100, 90, 80
SQL Server 2008	10	100	100, 90, 80
SQL Server 2005 (9.x)	9	90	90, 80
SQL Server 2000 (8.x)	8	80	80

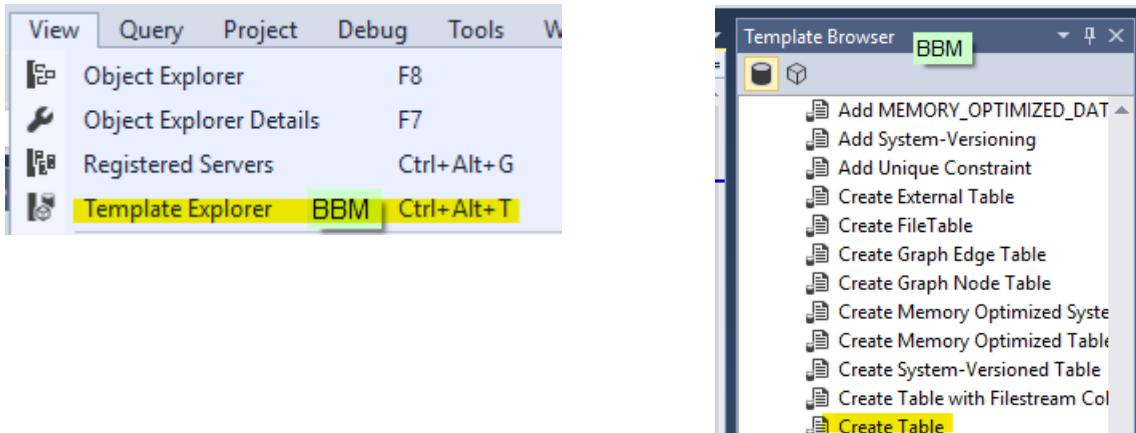
Consultamos el nivel de compatibilidad de nuestra base y vemos que es **140**. Para cambiarlo, usamos el comando **ALTER DATABASE CURRENT SET COMPATIBILITY_LEVEL = 130** acompañado de **ALTER DATABASE CURRENT SET MEMORY_OPTIMIZED_ELEVATE_TO_SNAPSHOT = ON**.

```
/*OPTIMIZAR LA BASE DE DATOS*/
SELECT d.compatibility_level
    FROM sys.databases AS d
    WHERE d.name = Db_Name();
go
-- compatibility_level
-- 140

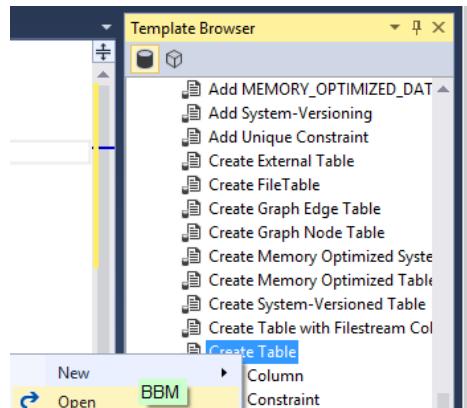
ALTER DATABASE CURRENT
    SET COMPATIBILITY_LEVEL = 130;
GO
ALTER DATABASE CURRENT
    SET MEMORY_OPTIMIZED_ELEVATE_TO_SNAPSHOT = ON;
GO

SELECT d.compatibility_level
    FROM sys.databases AS d
    WHERE d.name = Db_Name();
go
--compatibility_level
--130
```

Ahora voy a introducir una nueva manera de crear objetos, los **Templates**. Los *templates* son scripts de *SQL Server* que contienen código SQL, que frecuentemente se usan por los DBA para optimizar el trabajo. Si nos vamos a **View>Template Browser**, se abre un *Browser* donde podemos ver varios templates.



Hay tres maneras de insertar un *template SQL* en una *query*. La primera es haciendo doble clic o clic de botón derecho sobre una tabla del Template Explorer y darle a Open.



Como se ve, se abre una query nueva:

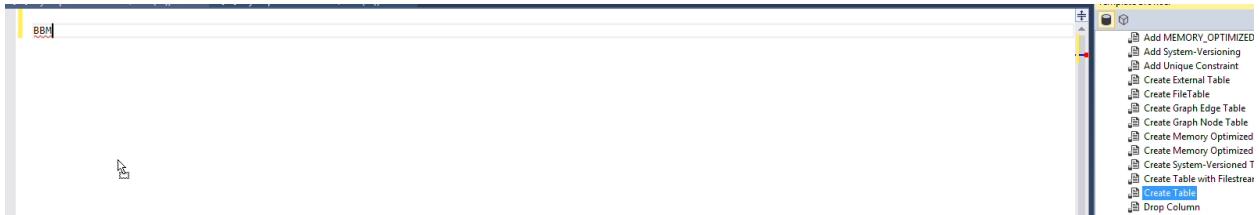
```
SQLQuery3.sql - D...MASPACE\BBM (53) -> X SQLQuery1.sql - D...MASPACE\BBM (56)*
-----[+]
-- Create table template
-----[+]
USE <database, sysname, AdventureWorks>
GO

IF OBJECT_ID('<schema_name, sysname, dbo>.<table_name, sysname, sample_table>', 'U') IS NOT NULL
    DROP TABLE <schema_name, sysname, dbo>.<table_name, sysname, sample_table>
GO

CREATE TABLE <schema_name, sysname, dbo>.<table_name, sysname, sample_table>
(
    <columns_in_primary_key, , c1> <column1_datatype, , int> <column1_nullability,, NOT NULL>,
    <column2_name, sysname, c2> <column2_datatype, , char(10)> <column2_nullability,, NULL>,
    <column3_name, sysname, c3> <column3_datatype, , datetime> <column3_nullability,, NULL>,
    CONSTRAINT <constraint_name, sysname, PK_sample_table> PRIMARY KEY (<columns_in_primary_key, , c1>)
)
GO
```

La tercera es un *drag and drop* desde el *Template Browser* hacia la *query* que se desea insertar.

- Drag and Drop



- Resultado

```

-- Create table template
-- <table_name>
USE <database>, sysname, AdventureWorks
GO

IF OBJECT_ID('<schema_name>, sysname, dbo>.<table_name>', 'U') IS NOT NULL
    DROP TABLE <schema_name>, sysname, dbo>.<table_name>
GO

CREATE TABLE <schema_name>, sysname, dbo>.<table_name>, sysname, sample_table>
(
    <columns_in_primary_key>, c1 <column1_datatype>, int <column1_nullability>, NOT NULL,
    <column2_name>, sysname, c2 <column2_datatype>, char(10) <column2_nullability>, NULL,
    <column3_name>, sysname, c3 <column3_datatype>, datetime <column3_nullability>, NULL,
    CONSTRAINT <constraint_name>, sysname, PK_sample_table> PRIMARY KEY (<columns_in_primary_key>, c1)
)
GO
  
```

Para el ejemplo que voy a utilizar, queda un script de esta manera:

```

USE <database>, sysname, AdventureWorks
GO

--Drop table if it already exists.
IF OBJECT_ID('<schema_name>, sysname,
dbo>.<table_name>, sysname, sample_memoryoptimizedtable>', 'U') IS NOT NULL
    DROP TABLE <schema_name>, sysname,
dbo>.<table_name>, sysname, sample_memoryoptimizedtable>
GO

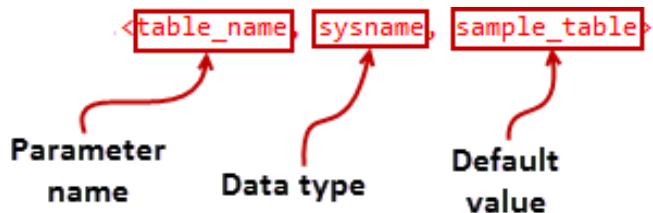
CREATE TABLE <schema_name>, sysname,
dbo>.<table_name>, sysname, sample_memoryoptimizedtable>
(
    <column_in_primary_key>, sysname, c1 <column1_datatype>, , int>
<column1_nullability>, , NOT NULL>,
    <column2_name>, sysname, c2 <column2_datatype>, , float>
<column2_nullability>, , NOT NULL>,
    <column3_name>, sysname, c3 <column3_datatype>, , decimal(10,2)>
<column3_nullability>, , NOT NULL> INDEX <index3_name>, sysname,
index_sample_memoryoptimizedtable_c3> NONCLUSTERED (<column3_name>,
sysname, c3>),

    CONSTRAINT <constraint_name>, sysname, PK_sample_memoryoptimizedtable>
PRIMARY KEY NONCLUSTERED (<column1_name>, sysname, c1>),
    -- See SQL Server Books Online for guidelines on determining
    appropriate bucket count for the index
  
```

```
INDEX <index2_name, sysname,
hash_index_sample_memoryoptimizedtable_c2> HASH <>column2_name, sysname,
c2>) WITH (BUCKET_COUNT = <sample_bucket_count, int, 131072>)
) WITH (MEMORY_OPTIMIZED = ON, DURABILITY = <durability_type, ,
SCHEMA_AND_DATA>)
GO
```

La plantilla SQL puede incluir o no parametrización. Los parámetros de la plantilla SQL son marcadores de posición para los valores que el usuario debe cambiar. Un parámetro de plantilla SQL comienza con un signo menor que "<" y termina con un signo mayor que ">". Hay tres partes del parámetro de plantilla SQL entre el signo menor que y mayor que:

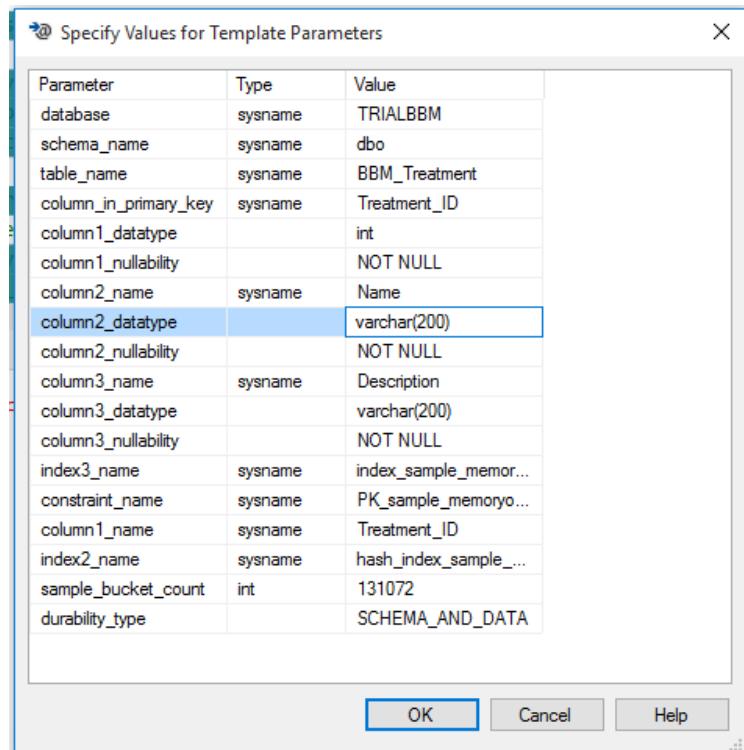
- Nombre del parámetro: el nombre del parámetro que debe cambiarse (por ejemplo, nombre_esquema, nombre_base_datos, nombre_tabla).
- Tipo de datos: el tipo de datos del parámetro (por ejemplo, int, varchar, date, sysname).
- Valor: el parámetro muestra qué valor se utilizará como predeterminado para cada parámetro.



El tipo de datos y el valor son opcionales y se pueden omitir de la lista de parámetros. Tenga en cuenta que cuando se omite, la coma que los separa debe permanecer:

```
<table_name, ,>
```

Una vez dicho esto, ya sólo queda especificar los valores. Se puede hacer de manera gráfica o con el conjunto de teclas **Ctrl + Shift + M** (en este caso lo que he hecho yo). Sale un *pop-up* en el que se ven campos que se pueden cambiar, como el nombre de la tabla, *schema*, etc. Yo los cubrí así.



Una vez le doy a Ok, los datos que antes estaban entre <> se cubren automáticamente quedando de la siguiente manera. Primero como siempre, controlo la existencia de la tabla:

```
USE TRIALBBM
GO

--Drop table if it already exists.
IF OBJECT_ID('dbo.BBM_Treatment','U') IS NOT NULL
    DROP TABLE dbo.BBM_Treatment
GO

CREATE TABLE dbo.BBM_Treatment
(
    Treatment_ID int NOT NULL,
    Name varchar(200) NOT NULL,
    Description varchar(200) NOT NULL INDEX
    index_sample_memoryoptimizedtable_c3 NONCLUSTERED (Description),

    CONSTRAINT PK_sample_memoryoptimizedtable PRIMARY KEY NONCLUSTERED
    (Treatment_ID),
    -- See SQL Server Books Online for guidelines on determining
    appropriate bucket count for the index
    INDEX hash_index_sample_memoryoptimizedtable_c2 HASH (Name) WITH
    (BUCKET_COUNT = 131072)
) WITH (MEMORY_OPTIMIZED = ON, DURABILITY = SCHEMA_AND_DATA)
GO
```

Entre las opciones para poner en las *tablas in memory*, vemos que hay un componente que es durability. Esto lleva a que hay dos tipos de tablas *in-memory OLTP*: Las **durable** y las **non-durable**.

Hay situaciones en las que los datos son temporales y está bien para perderlos debido al reinicio de SQL Server, entonces se puede pensar en tablas no duraderas en OLTP en memoria de SQL Server. De forma predeterminada, las tablas son duraderas. Esto se define mediante el parámetro **DURABILITY** durante la sintaxis de la tabla creada. Esto indica si queremos que los datos de la tabla se conserven cuando se reinicia la base de datos (a través de off line / online, reinicio del servicio SQL o bloqueo de SQL). Las tablas no duraderas tendrían **SCHEMA_ONLY** como valor pasado al parámetro de durabilidad, y las duraderas el parámetro **SCHEMA_AND_DATA**.

NOTA: Hay que tener en cuenta que estas tablas son diferentes de la tabla temporal que se crea en *tempdb* (#table y ## table). Una diferencia importante es que las tablas *tempdb* no están optimizadas para la memoria y no podrán sobrevivir después del reinicio, incluido el esquema.

Con esta información, controlamos la existencia de una segunda tabla y creamos la tabla **BBM_User**:

```
USE TRIALBBM
GO

DROP TABLE IF EXISTS BBM_User
GO
Create Table BBM_User
(
User_ID INT NOT NULL Primary key nonclustered Hash WITH (bucket_Count =
1000000),
FirstName_1 Nvarchar(20) NOT NULL,
FirstName_2 Nvarchar(20) NOT NULL,
Lastname_1 Nvarchar(20)NOT NULL,
Lastname_2 Nvarchar(20)NOT NULL,
DNI Nvarchar(9) NOT NULL,
Other_Details Nvarchar(20) NOT NULL,
)
With(Memory_optimized=on,Durability=SCHEMA_And_DATA)
GO
```

Ahora vamos a insertar los valores. El comando **SET nocount ON** ayuda a evitar que se devuelva el mensaje que muestra el recuento del número de filas afectadas por una instrucción o un procedimiento almacenado. Hacemos que en la introducción de datos se declare una variable **@counter** con el comando **DECLARE**, que aumenta lo que le mandemos mientras (**WHILE**) sea menor o igual a 100000.

```
-- BBM_User
SET nocount ON
GO
```

```

DECLARE @counter INT
SET @counter = 1
WHILE @counter <= 100000
BEGIN
    INSERT INTO dbo.BBM_User
    VALUES (@counter, 'Paco', 'Diego', 'Maradona', 'León', '12345678C', 'DETALLES
1'),
    (@counter+1,
    'Teresa', 'Carmen', 'Galilei', 'Einstein', '12345679K', 'DETALLES 2'),
    (@counter+2, 'Pepe', 'El', 'Gran', 'Viyuela', '12345678T', 'DETALLES 3')

SET @counter = @counter + 3
END
GO

SELECT * FROM BBM_User
GO

```

User_ID	FirstName_1	FirstName_2	LastName_1	LastName_2	DNI	Other_Details
301	99266	Teresa	Carmen	Galilei	Einstein	12345679K DETALLES 2
302	99268	Paco	Diego	Maradona	León	12345678C DETALLES 1
303	99269	Teresa	Carmen	Galilei	Einstein	12345679K DETALLES 2
304	99271	Paco	Diego	Maradona	León	12345678C DETALLES 1
305	99272	Teresa	Carmen	Galilei	Einstein	12345679K DETALLES 2
306	99274	Paco	Diego	Maradona	León	12345678C DETALLES 1
307	99275	Teresa	Carmen	Galilei	Einstein	12345679K DETALLES 2
308	99277	Paco	Diego	Maradona	León	12345678C DETALLES 1

Inserto ahora los tratamientos. Me va a dar un error por la **falta de Hardware** en mi máquina virtual:

```

-- BBM_Treatment
set nocount on
go
DECLARE @counter INT
SET @counter = 1
WHILE @counter <= 100000
BEGIN
    INSERT INTO dbo.BBM_Treatment
    VALUES (@counter, 'Esquizofrenia', 'ESTO ES LA DESCRIPCIÓN DEL
TRATAMIENTO UNO'),
    (@counter+1, 'TDAH', 'ESTO ES LA DESCRIPCIÓN DEL TRATAMIENTO DOS'),
    (@counter+2, 'TOC', 'ESTO ES LA DESCRIPCIÓN DEL TRATAMIENTO TRES')

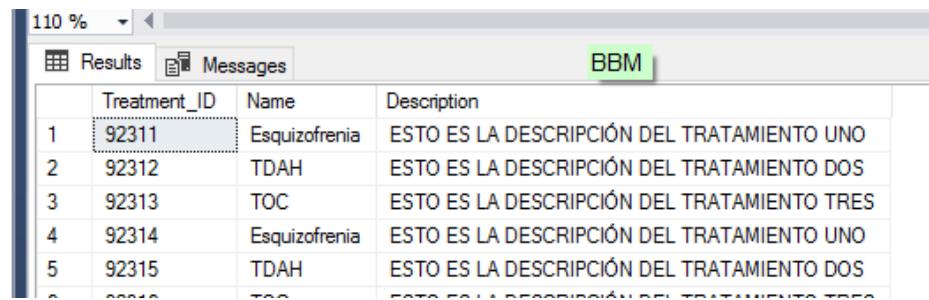
SET @counter = @counter + 3
END
GO

--The statement has been terminated.
--Msg 701, Level 17, State 109, Line 183
--There is insufficient system memory in resource pool 'default' to run
this query.

--FASTER

SELECT * FROM BBM_Treatment
GO

```



	Treatment_ID	Name	Description
1	92311	Esquizofrenia	ESTO ES LA DESCRIPCIÓN DEL TRATAMIENTO UNO
2	92312	TDAH	ESTO ES LA DESCRIPCIÓN DEL TRATAMIENTO DOS
3	92313	TOC	ESTO ES LA DESCRIPCIÓN DEL TRATAMIENTO TRES
4	92314	Esquizofrenia	ESTO ES LA DESCRIPCIÓN DEL TRATAMIENTO UNO
5	92315	TDAH	ESTO ES LA DESCRIPCIÓN DEL TRATAMIENTO DOS

Ahora dejo los comandos de como fuerzo a mi base de datos a trabajar OFFLINE (la base de datos está cerrada, se ha cerrado correctamente y se ha marcado como sin conexión. La base de datos no se puede modificar mientras está desconectada) u ONLINE (la base de datos está abierta y disponible para su uso).

```
USE Master
go
ALTER DATABASE TRIALBBM SET OFFLINE WITH ROLLBACK IMMEDIATE
GO
ALTER DATABASE TRIALBBM SET ONLINE
GO
```

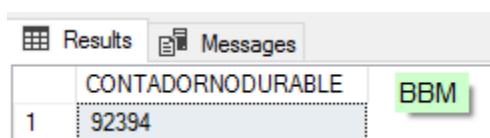
Si verificamos ahora el recuento de datos en ambas tablas que encontrará No hay datos en no duradero en la tabla de memoria. Por lo tanto, al final, hay que tener cuidado con la tabla duradera y la tabla no duradera en la memoria. La tabla optimizada debe saber cuándo necesita usar esto o esto. Si no importa la pérdida de datos, se puede usar la tabla no duradera en la memoria.

```
USE [TRIALBBM]
GO

SELECT * FROM BBM_User
GO

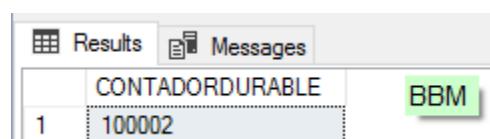
SELECT * FROM BBM_Treatment
GO

Select Count(1) AS CONTADORNODURABLE from BBM_Treatment
GO
```



	CONTADORNODURABLE
1	92394

```
Select Count(1) AS CONTADORDURABLE from BBM_User
GO
```



	CONTADORDURABLE
1	100002

BACKUP ALL CURSORS

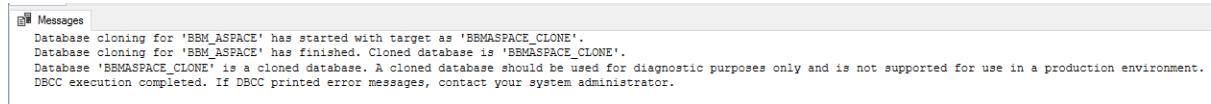
A veces, las cosas que parecen complicadas son mucho más fáciles de lo que piensas y este es el poder de usar T-SQL para encargarse de tareas repetitivas. Una de estas tareas puede ser la necesidad de realizar una copia de seguridad de todas las bases de datos de su servidor. Con el uso de T-SQL se pueden generar a través de comandos de respaldo y con el uso de cursosres puede desplazarse por todas las bases de datos para respaldarlas una por una. Este es un proceso muy sencillo y solo necesita un puñado de comandos para hacer esto.

Para este script voy a hacer las pruebas como siempre de aislar el caso en una base de datos aparte, probarlo, y luego sería incorporarlo a la base de datos que tengo.

Para este fin utilizo el comando **DBCC CLONESATABASE (<base_datos_origen>, <base_datos_destino>)**.

```
USE BBM_ASPACE
GO

/*CLONO BASE PARA HACER LAS PRUEBAS*/
DBCC CLONEDATABASE (BBM_ASPACE, BBMASPACE_CLONE);
GO
```

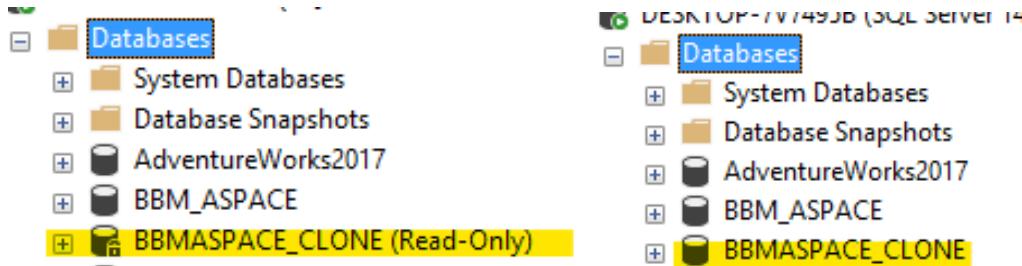


Database cloning for 'BBM_ASPACE' has started with target as 'BBMASPACE_CLONE'.
 Database cloning for 'BBM_ASPACE' has finished. Cloned database is 'BBMASPACE_CLONE'.
 Database 'BBMASPACE_CLONE' is a cloned database. A cloned database should be used for diagnostic purposes only and is not supported for use in a production environment.
 DBCC execution completed. If DBCC printed error messages, contact your system administrator.

Un dato curioso, es que la base de datos que se clona, lo hace sólo en Read-Only, para ello, utilizo el comando **ALTER DATABASE <base_datos> SET READ-WRITE**.

```
USE BBMASPACE_CLONE
GO

-- POR DEFECTO EL CLONADO LO CREA EN READ ONLY, ASÍ QUE YO LO PONGO EN
-- READ/WRITE
ALTER DATABASE BBMASPACE_CLONE SET READ_WRITE
GO
```



Ahora para la prueba, ¿Cómo hago el *backup* de las bases de datos? Pues:

Me coloco en la base de datos master y creo un procedimiento llamado BBM_CURSORBACKUP donde:

- Declaro cuatro variables:
 - Nombre de la base de datos
 - Ruta para los backups
 - El nombre del archivo
 - La fecha con formato aaddmm.bak.

```
USE master
GO
CREATE OR ALTER PROC BBM_CURSORBACKUP
AS
  BEGIN
    DECLARE @name VARCHAR(50) -- database name
    DECLARE @path VARCHAR(256) -- path for backup files
    DECLARE @fileName VARCHAR(256) -- filename for backup
    DECLARE @fileDate VARCHAR(20) -- used for file name
```

Especifico el directorio de la base de datos (SET)

```
-- specify database backup directory
SET @path = 'C:\BBM_BACKUPS\'
```

Especifico el formato de la fecha del archivo con un SELECT de la fecha

```
SELECT @fileDate = CONVERT(VARCHAR(20),GETDATE() ,112) +
REPLACE(CONVERT(VARCHAR(20),GETDATE() ,108),':','')
  DECLARE db_cursor CURSOR READ_ONLY FOR
  SELECT name
  FROM master.dbo.sysdatabases
  WHERE name IN ('BBMASPACE_CLONE')
```

Le digo que el cursor apunte al nombre e indico que la variable nombre de archivo será la ruta + _ + la fecha + .bak

```
OPEN db_cursor

FETCH NEXT FROM db_cursor INTO @name

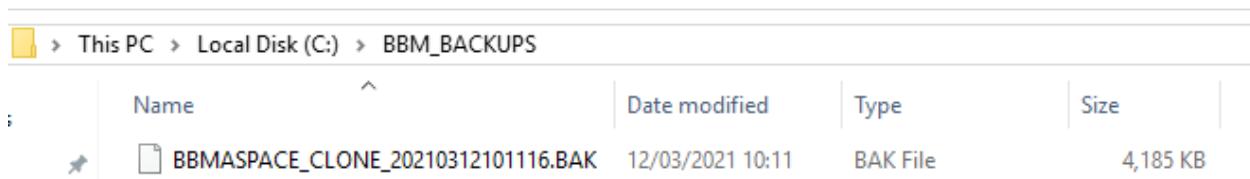
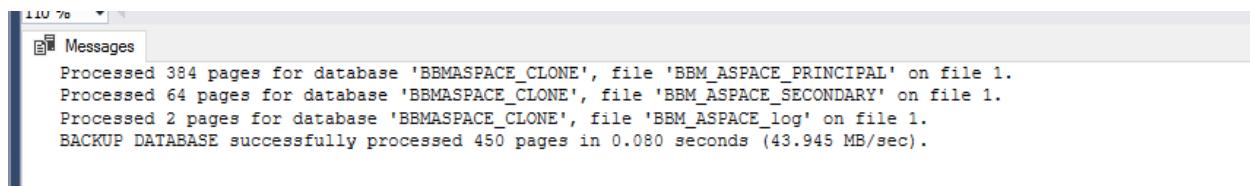
WHILE @@FETCH_STATUS = 0
  BEGIN
```

```
SET @fileName = @path + @name + '_' + @fileDate + '.BAK'
BACKUP DATABASE @name TO DISK = @fileName

FETCH NEXT FROM db_cursor INTO @name
END
CLOSE db_cursor
DEALLOCATE db_cursor
END
```

Ejecuto el procedimiento almacenado que acabo de crear:

```
EXECUTE BBM_CURSORBACKUP
GO
```



Altero la base con **ALTER DATABASE <base_datos> SET READ-ONLY** para ponerla en Read-Only y la borro con **DROP**.

```
ALTER DATABASE BBMASPACE_CLONE SET READ_ONLY
GO

DROP DATABASE BBMASPACE_CLONE
GO
```

STORED PROCEDURE BACKUP

Un procedimiento almacenado de SQL Server **es un grupo de una o más instrucciones Transact-SQL**. Los procedimientos se asemejan a las construcciones de otros lenguajes de programación, porque pueden:

- Aceptar parámetros de entrada y devolver varios valores en forma de parámetros de salida al programa que realiza la llamada.
- Contener instrucciones de programación que realicen operaciones en la base de datos. Entre otras, pueden contener llamadas a otros procedimientos.
- Devolver un valor de estado a un programa que realiza una llamada para indicar si la operación se ha realizado correctamente o se han producido errores, y el motivo de estos.

Hay varios tipos de procedimientos almacenados:

- **Definidos por el usuario** → se puede crear en una base de datos definida por el usuario o en todas las bases de datos del sistema excepto en la base de datos Resource.
- **Temporales** → son una forma de procedimientos definidos por el usuario. Los procedimientos temporales son iguales que los procedimientos permanentes salvo porque se almacenan en tempdb. Hay dos tipos de procedimientos temporales, se diferencian entre sí por los nombres, la visibilidad y la disponibilidad:
 - **Locales** → tienen como primer carácter de sus nombres un solo signo de número (#); solo son visibles en la conexión actual del usuario y se eliminan cuando se cierra la conexión.
 - **Globales** → presentan dos signos de número (##) antes del nombre; son visibles para cualquier usuario después de su creación y se eliminan al final de la última sesión en la que se usa el procedimiento.
- **Sistema** → se incluyen con SQL Server. Están almacenados físicamente en la base de datos interna y oculta **Resource** y se muestran de forma lógica en el esquema **sys** de cada base de datos definida por el sistema y por el usuario. Además, la base de datos **msdb** también contiene procedimientos almacenados del sistema en el esquema **dbo** que se usan para programar alertas y trabajos. Dado que los procedimientos del sistema empiezan con el prefijo **sp_SQL** Server admite los procedimientos del sistema que proporcionan una interfaz de SQL Server a los programas externos para varias actividades de mantenimiento. Estos procedimientos extendidos usan el prefijo **xp_**
- **Extendidos definidos por el usuario** → los procedimientos extendidos le permiten crear sus propias rutinas externas en un lenguaje de programación como puede

ser C. Estos procedimientos son DLL que una instancia de SQL Server puede cargar y ejecutar dinámicamente.

La cláusula **EXECUTE AS** puede especificarse en la instrucción **CREATE PROCEDURE** para habilitar la suplantación de otro usuario o para permitir que los usuarios o las aplicaciones puedan realizar ciertas actividades en la base de datos sin necesidad de contar con permisos directos sobre los objetos y comandos subyacentes.

Para este caso concreto voy a hacer un procedimiento almacenado creado por mí, lo que lo convierte en uno **temporal local definido por el usuario**.

Comienzo el proyecto haciendo lo mismo que en el ejemplo anterior, un clonado de la base de datos para una posterior implementación.

```
USE BBM_ASPACE
GO
```

```
/*CLONO BASE PARA HACER LAS PRUEBAS*/
DBCC CLONEDATABASE (BBM_ASPACE, BBMASPACE_CLONE);
GO
```

```
Messages
Database cloning for 'BBM_ASPACE' has started with target as 'BBMASPACE_CLONE'.
Database cloning for 'BBM_ASPACE' has finished. Cloned database is 'BBMASPACE_CLONE'.
Database 'BBMASPACE_CLONE' is a cloned database. A cloned database should be used for diagnostic purposes only and is not supported for use in a production environment.
DBCC execution completed. If DBCC printed error messages, contact your system administrator.
```

```
USE BBMASPACE_CLONE
GO
```

```
-- POR DEFECTO EL CLONADO LO CREA EN READ ONLY, ASÍ QUE YO LO PONGO EN
READ/WRITE
```

```
ALTER DATABASE BBMASPACE_CLONE SET READ_WRITE
GO
```

Para crear el procedimiento uso la sentencia **CREATE PROC** (Procedure) **<nombre_procedimiento>**. Marco la ruta también con una variable:

En el cuerpo AS meto:

- Una declaración de variables del **nombre de la base**, **nombre del archivo de backup**, la **fecha** y una variable *integer* **backupcount**.
- La creación de una tabla temporal **BBM_TEMPBACKUP**.
- Con **SET**, le doy formato a la fecha.
- Inserto valores en la tabla temporal

- Le meto un **SELECT** para ordenar
- También incluyo un condicional donde el contador del *backup* no es nulo y el contador es mayor a cero se cumple que:
 - Se declare que la variable *currentbackup* sea un *integer* e igual a 1 mientras que el *backup* actual sea menor o igual que el contador de *backup*
 - Se marcan los valores de las variables que dije al principio

```
USE master

GO
DROP PROCEDURE IF EXISTS BBM_SPBACKUP
GO
-- CREATE OR ALTER
CREATE PROC BBM_SPBACKUP
    @path VARCHAR(256)
AS
-- Declarando variables
DECLARE @name VARCHAR(50), -- database name
-- @path VARCHAR(256), -- path for backup files
@fileName VARCHAR(256), -- filename for backup
@fileDate VARCHAR(20), -- used for file name
@backupCount INT
CREATE TABLE [dbo].#BBM_TEMPBACKUP
(intID INT IDENTITY (1, 1),
name VARCHAR(200))

-- SET @path = 'C:\BBM BACKUPS\' 
-- Includes the date in the filename
SET @fileDate = CONVERT(VARCHAR(20), GETDATE(), 112)
-- Includes the date and time in the filename
--SET @fileDate = CONVERT(VARCHAR(20), GETDATE(), 112) + '_' +
REPLACE(CONVERT(VARCHAR(20), GETDATE(), 108), ':', '')
INSERT INTO [dbo].#BBM_TEMPBACKUP (name)
    SELECT name
    FROM master.dbo.sysdatabases
    WHERE name IN ('BBMASPACE_CLONE')
-- WHERE name NOT IN ('master', 'model', 'msdb', 'tempdb')

SELECT TOP 1 @backupCount = intID
FROM [dbo].#BBM_TEMPBACKUP
ORDER BY intID DESC
-- Utilidad: Solo Comprobación N° Backups a realizar
print @backupCount
IF (@backupCount IS NOT NULL) AND (@backupCount > 0))
BEGIN
    DECLARE @currentBackup INT
    SET @currentBackup = 1
    WHILE (@currentBackup <= @backupCount)
```

```

BEGIN
    SELECT
        @name = name,
        @fileName = @path + name + '_' + @fileDate + '.BAK'
    FROM [dbo].#BBM_TEMPBACKUP
    WHERE intID = @currentBackup

    -- Utilidad: Solo Comprobación Nombre Backup
    print @fileName
    -- does not overwrite the existing file
    BACKUP DATABASE @name TO DISK = @fileName
    --BACKUP DATABASE @name TO DISK = @fileName WITH INIT
    SET @currentBackup = @currentBackup + 1
END
GO

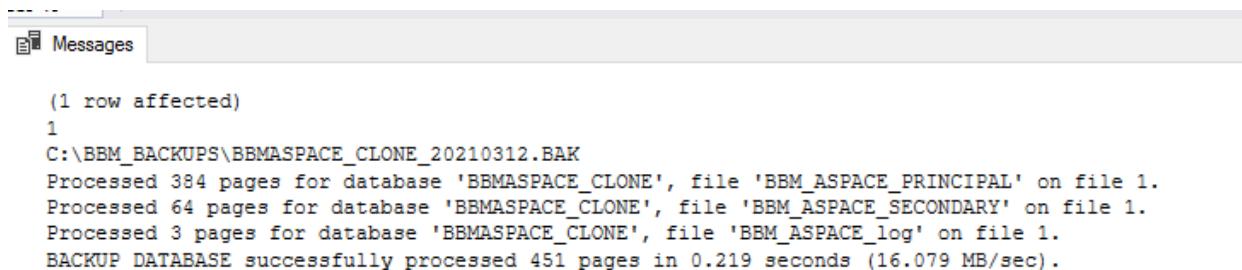
```

Ejecuto el procedimiento

```

-- Ejecutar Procedimiento
-- Input Parameter 'C:\Backup\' 
EXEC BBM_SPBACKUP 'C:\BBM_BACKUPS\' 
GO

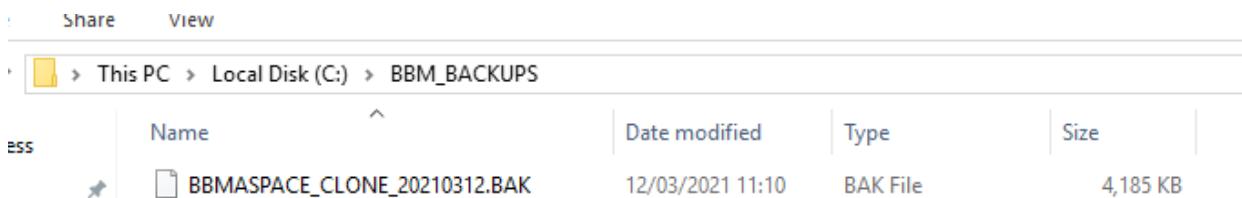
```



```

(1 row affected)
1
C:\BBM_BACKUPS\BBMASPACE_CLONE_20210312.BAK
Processed 384 pages for database 'BBMASPACE_CLONE', file 'BBM_ASPACE_PRINCIPAL' on file 1.
Processed 64 pages for database 'BBMASPACE_CLONE', file 'BBM_ASPACE_SECONDARY' on file 1.
Processed 3 pages for database 'BBMASPACE_CLONE', file 'BBM_ASPACE_log' on file 1.
BACKUP DATABASE successfully processed 451 pages in 0.219 seconds (16.079 MB/sec).

```



File	Name	Date modified	Type	Size
	BBMASPACE_CLONE_20210312.BAK	12/03/2021 11:10	BAK File	4,185 KB

Vuelvo la base Read-Only y borro la base de datos

```

ALTER DATABASE BBMASPACE_CLONE SET READ_ONLY
GO

DROP DATABASE BBMASPACE_CLONE
GO

```

SISTEMA DE CONTROL DE VERSIONES

¿Qué es un control de versiones? Un control de versiones es un sistema que registra los cambios realizados en un archivo o conjunto de archivos a lo largo del tiempo, de modo que puedas recuperar versiones específicas más adelante. Aunque en los ejemplos de este libro usarás archivos de código fuente como aquellos cuya versión está siendo controlada, en realidad puedes hacer lo mismo con casi cualquier tipo de archivo que encuentres en una computadora.

GIT como sistema de control de versiones distribuido

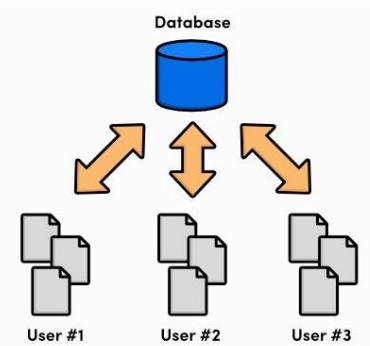
La finalidad que tiene es doble o triple:

- Organización de código. Repositorio de almacenaje de código.
- Controlar versiones. Se puede volver atrás en el código, deshaciendo cambios y volviendo atrás.
- Trabajo en equipo. Mantener varias personas usando el mismo código.

Antes había otros sistemas de control de versiones:

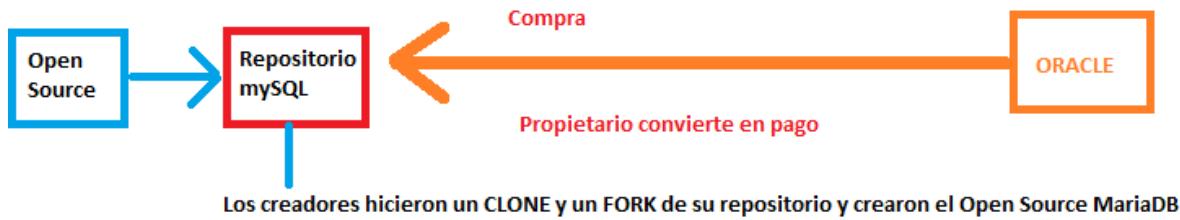
- Subversion
- Mercurial
- Team Foundation Server (TFS)

Todos estos eran sistemas de **control de versiones centralizados** (*SCV centralizados*), hasta que llegó Linux con **GIT**. La manera de trabajar en un SCV centralizado era, hay un *repositorio centralizado* y cada uno trabaja en su repositorio, y luego se cargaba en el centralizado.



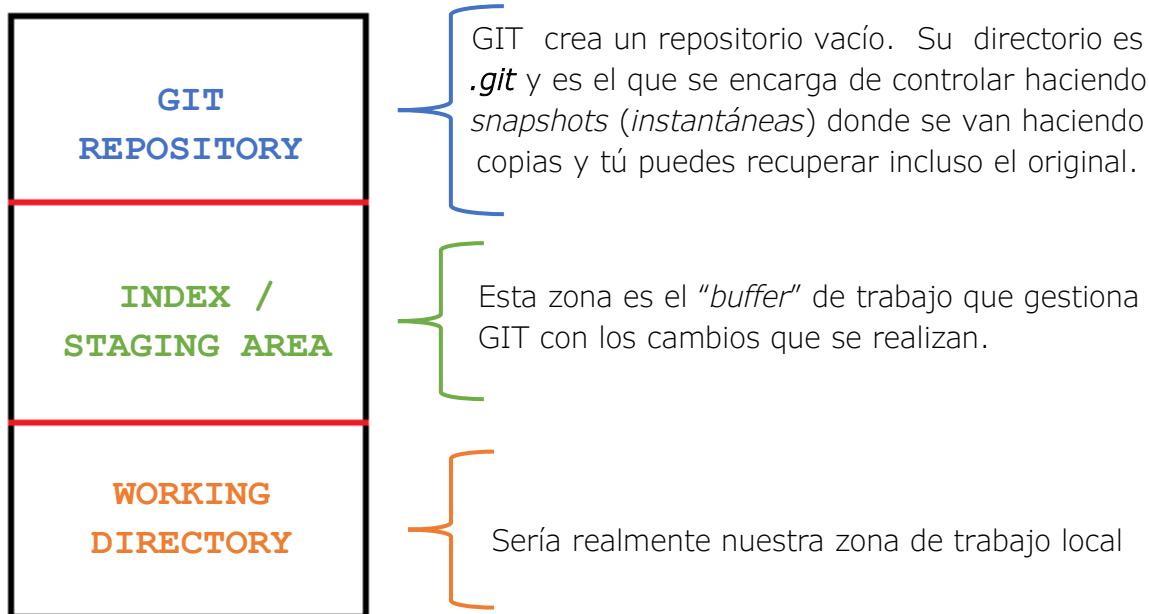
GIT es un sistema de **control de versiones distribuido** en el cual hay un *repositorio central*, y yo trabajo en mi desarrollador. En él puedo descargarme el repositorio central clonándolo (**clone**), para trabajar en el local con el repositorio central. Con este sistema, también puedo desarrollar mi propio código (haciendo lo que se llama **fork**). **Fork** significa *bifurcación*, donde yo puedo tener a mayores del repositorio central, un repositorio de desarrollo diferente del central.

Cuando un desarrollador o grupo quiere seguir otra línea de desarrollo diferente del repositorio central, tiene que hacer una **FORK** y continuar ahí en su desarrollo que nunca más se mezclará con el repositorio central origen. (por ejemplo *MariaDB* es un *fork del pasado sacado a partir del repositorio de MySQL*):



Sentencias GIT

Las principales **sentencias** de GIT se ven de la siguiente manera: yo tengo mi **directorio (repositorio)** y se distinguen dos áreas claras:



Sentencias GIT REPOSITORY

- **git ini** → crea un repositorio de GIT vacío que gestionará el propio GIT.

Sentencias INDEX / STAGING AREA

- **git status** → Trackea (sigue) todos los ficheros que tenemos en nuestro repositorio.
- **git add** → El comando para pasar del *Working Directory* al *Index / Staging* es puedo hacerlo de varias maneras, con un `.` me agrega todo.
- **git add nombre_fichero** → La otra manera de pasar del *Working Directory* al *Index Area* es así, moviendo lo que interesa.
- **git commit** → “Mueves” (confirma) pasar de *Index / Staging* a *Git Repository*. Un **commit** es técnicamente un *snapshot* (*instantánea*) de cómo está mi directorio de trabajo. Si yo sigo avanzando, siempre puedo volver atrás.
- **git diff** → Siver para ver los cambios que he hecho, así puedo comparar lo que hice un jueves con lo que hice el martes (por ejemplo).

Para obtener información de diferentes maneras tenemos:

- **git log** → Saca información de los *commit*.
- **git show** → Saca información de uno o más objetos.
- **git reflog** → Saca información cuando los tipos de ramas y otras referencias son actualizados en el repositorio local.

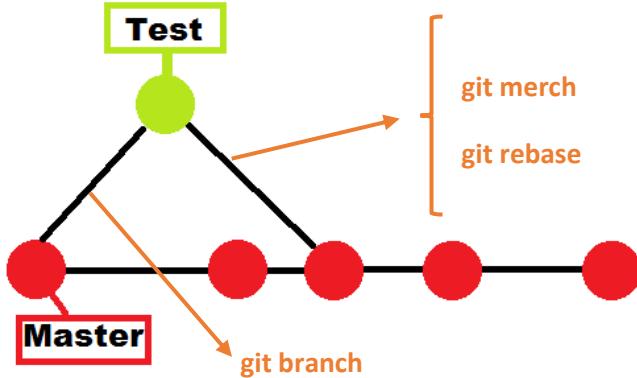
Los comandos para volver a momentos pasados son:

- **git reset** → Mueve el *HEAD* (un puntero que apunta a los *commit*) a un estado específico
- **git revert** → Revierte los *commit* existentes.
- **git checkout** → Son para volver a la versión anterior o cambiar entre *ramas*.

Las ramas (**branches**) son un apuntador móvil apuntando a una de esas confirmaciones. La rama principal se llama **máster** (desde octubre de 2020 se llama **main**) y es donde trabajamos habitualmente. Con la primera confirmación de cambios que realicemos, se creará esta rama principal máster apuntando a dicha confirmación. En cada confirmación de cambios que realicemos, la rama irá avanzando automáticamente. Como mencioné en el apartado anterior, te puedes mover entre ramas con el comando **git checkout**, pero... ¿y si queremos hacer otras acciones?

```
graph TD; Test((Test)) --- Master((Master)); Test --- Bottom(( )); Master --- Bottom;
```

git merch
git rebase
git branch



Para ello empleamos los siguientes comandos:

- **git branch rama_no_principal** → Sirve para crear otras ramas.
 - **git merch** → Sirve para unir *rama_no_principal* con la rama main (o actual).
 - **git rebase** → Reaplica *commits* sobre otra rama, haciendo cualquier actuación antes que cualquier otra cosa
 - **git stash** → Guarda los cambios en un *Working Directory* sucio

Actualmente todo el mundo trabaja en la **nube** (*cloud*) para hacerlo de manera **remota** (trabajo en equipo remoto). Yo puedo tener un servidor de repositorios local, pero no está en la nube, está en mi empresa. ¿Qué ocurre? Hay tres empresas grandes que dan muy buen servicio y son baratas (*GitHub*, *Gitlab* y *Bitbucket*), en ellos hay repositorios de todo tipo (públicos y privados) y con ellos podemos trabajar completamente en remoto. Entonces, ¿Qué falta?, lo que nos falta es como subir código y bajar código de la nube (básicamente los comandos para trabajar en la nube:

- **git push** → Para subir la información a la nube
 - **git pull** → Para obtener el repositorio en local de la nube.
 - **git clone <url>** → Se descarga el repositorio en la nube a un directorio local.
 - **git checkout** → Para moverse entre las ramas

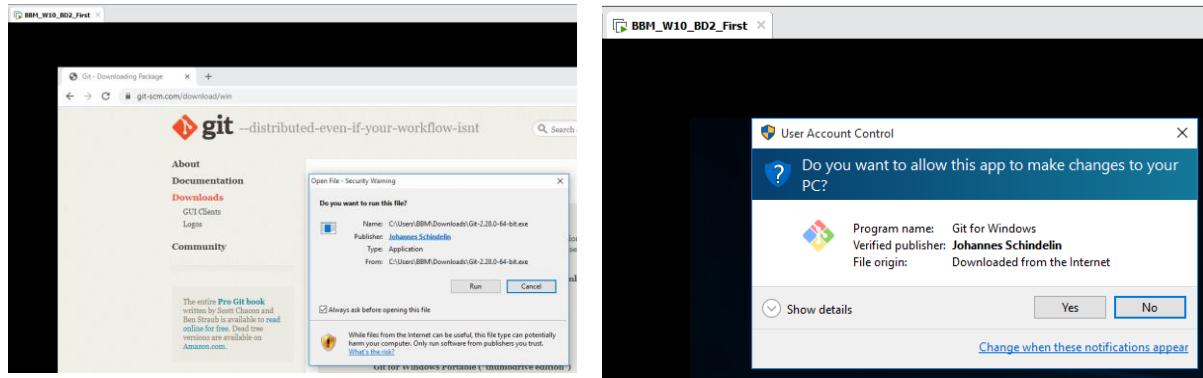
Para ignorar algunas partes del código utilizamos el comando

- **git ignore** → Sirve para que GIT ignore intencionadamente los archivos *untracked* (no seguidos).

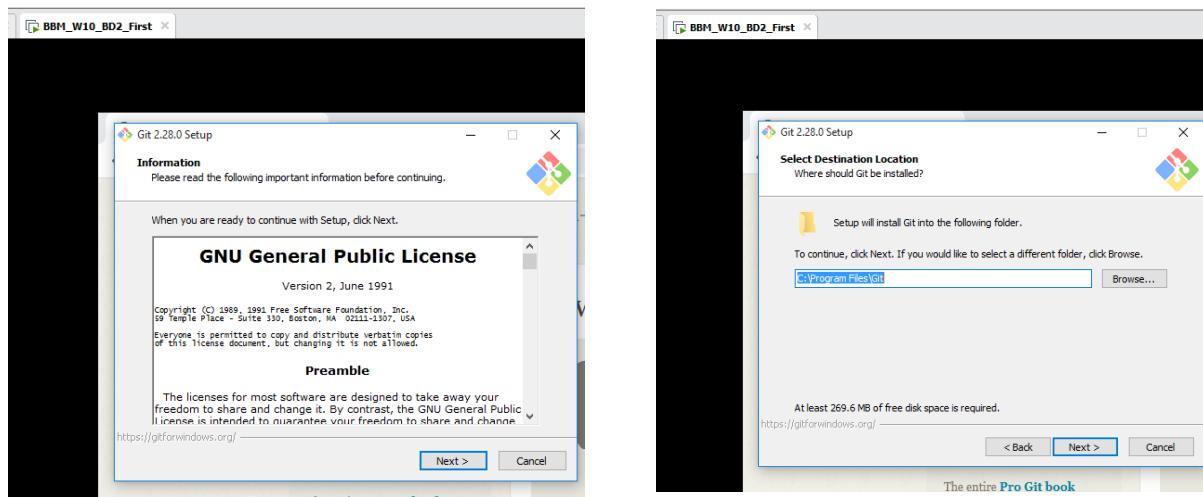
Instalando GIT

Instalación del "cmd" de GIT

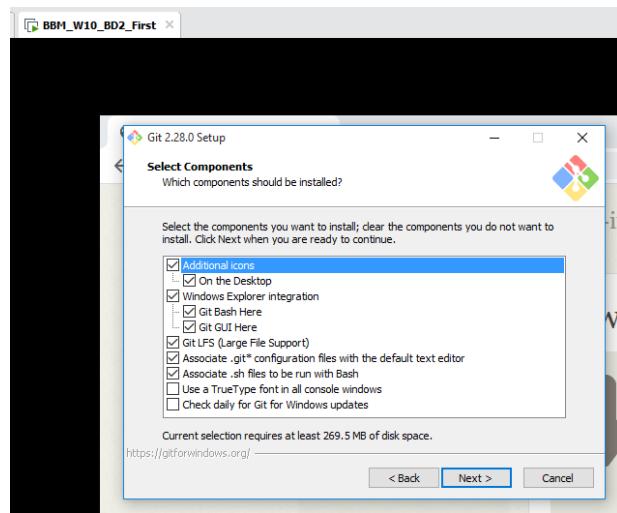
Descargamos la aplicación de la página <https://git-scm.com/download/win>. Elegimos la opción de Windows (versión 64-bit). Una vez descargado, lo ejecutamos como administrador.



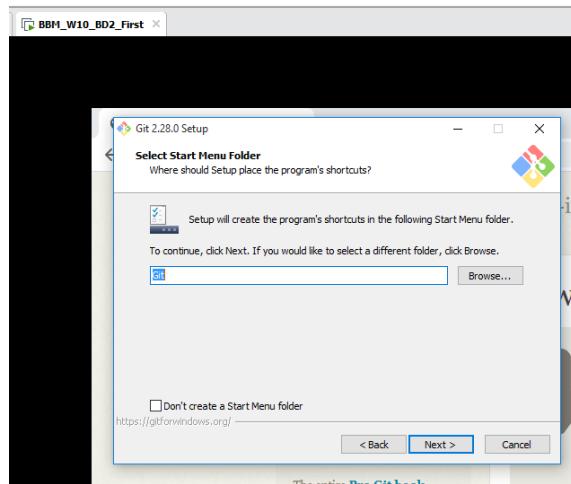
La verdad es que la instalación es muy genérica, dejamos las dos siguientes ventanas por defecto dándole a **Next**.



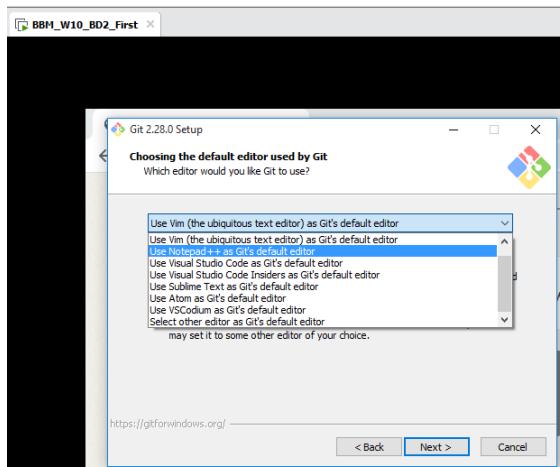
En la siguiente venta seleccionamos **Additional Icons** y hacemos clic en **Next**.



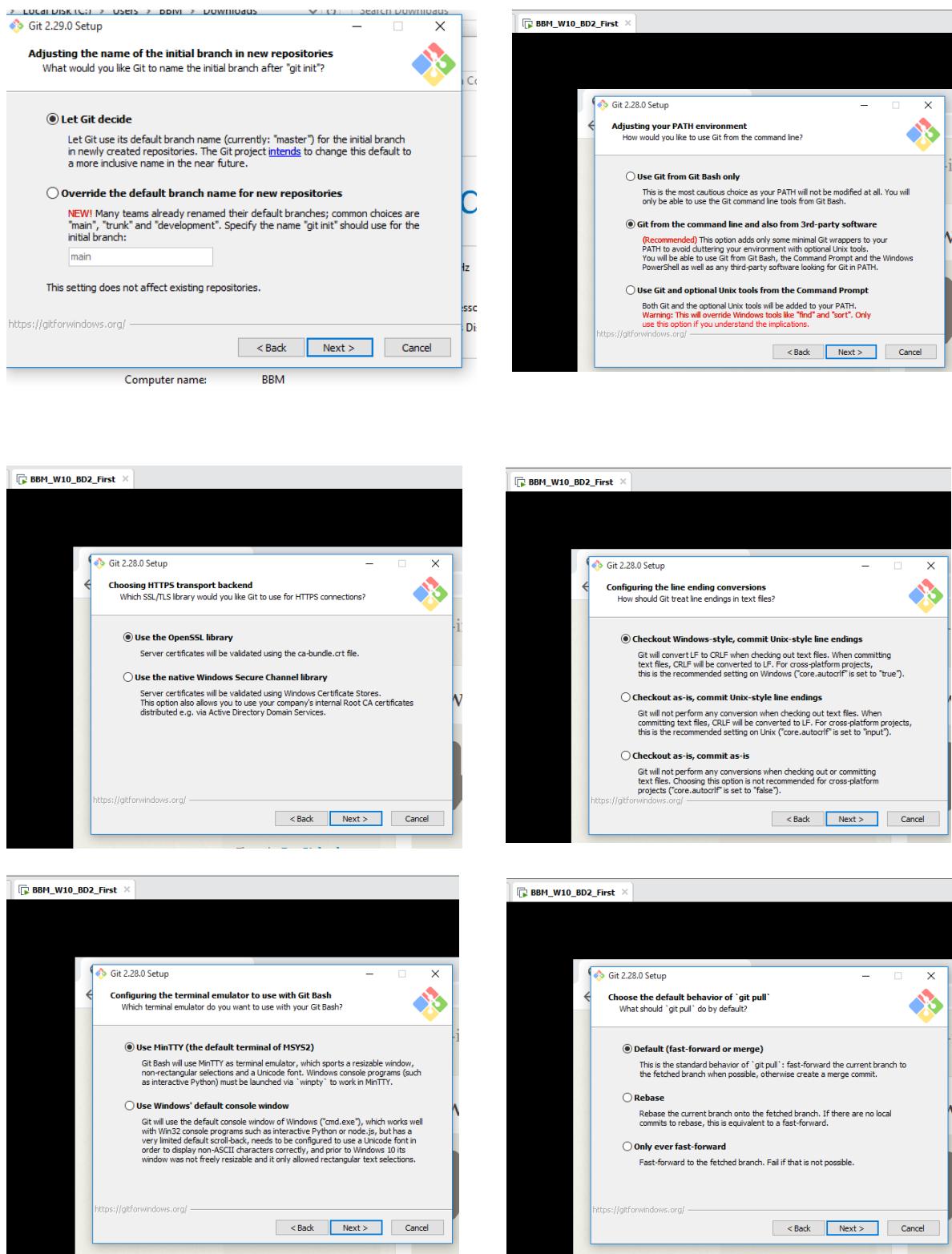
Dejamos esta ventana por defecto y clic en *Next*.



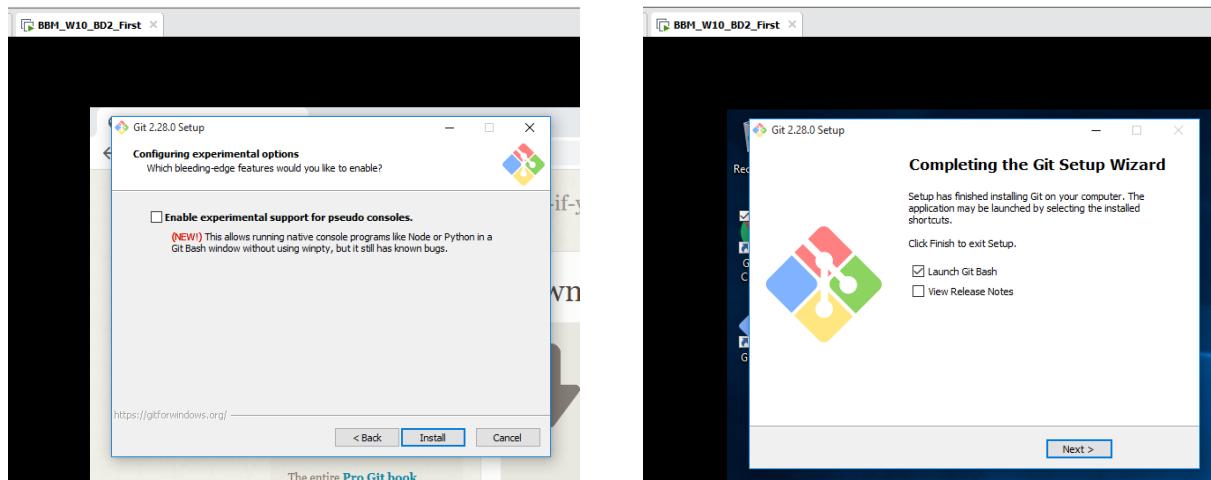
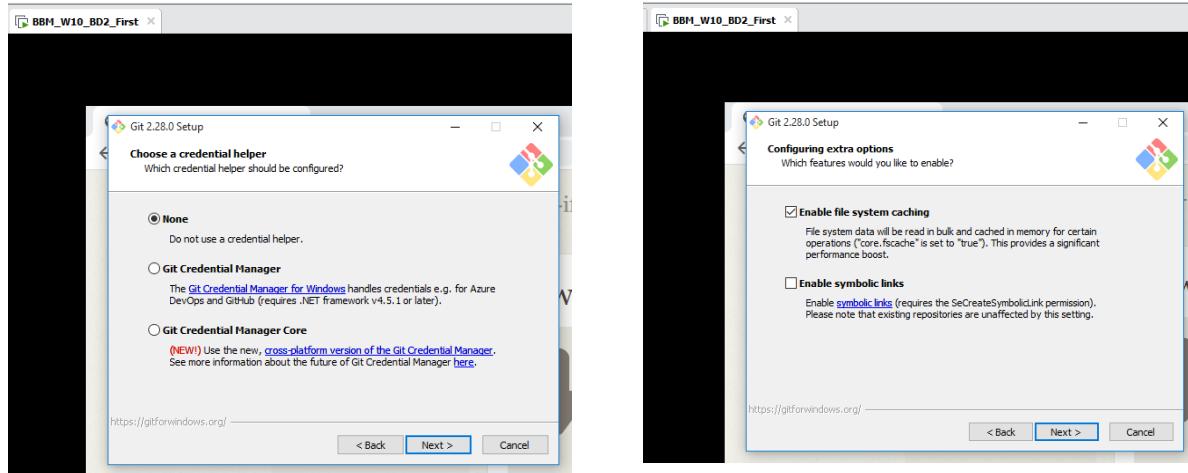
En la siguiente ventana tenemos que elegir un editor por defecto. En nuestro caso elegimos **Notepad ++**.



Desde aquí son todo ventanas por defecto y clicar en *Next*.



En esta ventana elegimos **None** como opción, y en la siguiente dejamos la opción por defecto hasta darle a **Install**. Una vez hecho esto, habremos acabado. Cuando esto suceda, iniciamos GIT.



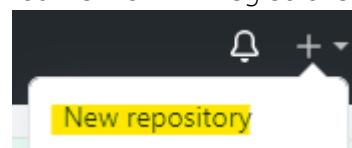
Entornos gráficos

En cuanto entornos gráficos que vamos a instalar, el primero será TORTOISE GIT (<https://tortoisegit.org/>) en Windows 10.

Primeros pasos con GIT

Crear un repositorio en GitHub

Desde GitHub creas un repositorio con el botón "+" (⊕ + ⚙) de arriba a la derecha. Obviamente tienes que haberte registrado en GitHub para comenzar. El registro es gratuito (Para este caso me creé una cuenta específica de *Gmail* llamada clase.BBM@gmail.com y la asocié a una cuenta de GitHub como usuario *BBMASAX*). Al hacer clic en el "+", pulsamos *New repository* y empezamos a crearlo.



Apareces en una página para que indiques los datos del repositorio. Básicamente tienes que darle un nombre, elegir si es público o privado y opcionalmente una descripción. En este caso marcamos la parte que pone *Add a README file*, porque se va a utilizar más adelante

Owner *
Repository name *

BBMASAX

/

Great repository names are short and memorable. Need inspiration? How about [literate-octo-journey](#)?

Description (optional)

Proyecto de evaluación Manuel Beade Boán

Public
 Anyone on the internet can see this repository. You choose who can commit.

Private
 You choose who can see and commit to this repository.

Initialize this repository with:

Skip this step if you're importing an existing repository.

Add a README file
 This is where you can write a long description for your project. [Learn more](#).

Add .gitignore
 Choose which files not to track from a list of templates. [Learn more](#).

Choose a license
 A license tells others what they can and can't do with your code. [Learn more](#).

Hacer el repositorio público o privado te lo tienen que decir las personas que te han pedido realizar esta tarea. En este caso, elegimos Público porque así es como se nos ha pedido, y le damos a **Create repository**.

Uso del Command Prompt de GIT

Todo el trabajo que se explique aquí será sobre un script en el que iré comentando los distintos sucesos que van ocurriendo.

Inicio con GIT Bash

```
BBM@DESKTOP-7V7495B MINGW64 ~/AppData/Local /* TE DICE CUÁL ES EL  
DIRECTORIO ACTUAL*/  
$ pwd  
/c/Users/BBM/AppData/Local  
  
BBM@DESKTOP-7V7495B MINGW64 ~/AppData/Local  
$ git version  
git version 2.29.0.windows.1  
  
BBM@DESKTOP-7V7495B MINGW64 ~/AppData/Local  
$ cd /c  
  
BBM@DESKTOP-7V7495B MINGW64 /c /* CREAR DIRECTORIO */  
$ mkdir prueba_git  
  
BBM@DESKTOP-7V7495B MINGW64 /c  
$ cd prueba_git  
  
BBM@DESKTOP-7V7495B MINGW64 /c/prueba_git  
$ git init  
Initialized empty Git repository in C:/prueba_git/.git/  
  
BBM@DESKTOP-7V7495B MINGW64 /c/prueba_git /* ASOCIAR USUARIO A  
REPOSITORIO */  
$ git config --global user.name beadebmanuel  
  
BBM@DESKTOP-7V7495B MINGW64 /c/prueba_git /* ASOCIAR EMAIL A REPOSITORIO  
*/  
$ git config --global user.email clase.bbm@gmail.com  
  
BBM@DESKTOP-7V7495B MINGW64 /c/prueba_git /* VER EL ESTADO DEL  
REPOSITORIO */  
$ git status  
On branch master  
  
No commits yet  
  
nothing to commit (create/copy files and use "git add" to track)  
  
BBM@DESKTOP-7V7495B MINGW64 /c/prueba_git /* COMPROBAR DIRECTORIO PARA  
COPIAR INSTPUBS.SQL */  
$ cd  
  
BBM@DESKTOP-7V7495B MINGW64 ~  
$ cd /c  
  
BBM@DESKTOP-7V7495B MINGW64 /c
```

```

$ ls -l
total 1704377
drwxr-xr-x 1 BBM 197121          0 Oct 20 11:24 '$Recycle.Bin'/
drwxr-xr-x 1 BBM 197121          0 Oct 22 10:29 '$WINDOWS.~BT'/
-r--r--r-- 1 BBM 197121          395268 Jul 10 2015 bootmgr
-rw-r--r-- 1 BBM 197121          1 Jul 10 2015 BOOTNXT
drwxr-xr-x 1 BBM 197121          0 Oct 22 10:45 Default_databases/
lrwxrwxrwx 1 BBM 197121          8 Jul 10 2015 'Documents and Settings'
-> /c/Users/
-rw-r--r-- 1 BBM 197121 1476395008 Oct 20 20:29 pagefile.sys
drwxr-xr-x 1 BBM 197121          0 Jul 10 2015 PerfLogs/
drwxr-xr-x 1 BBM 197121          0 Oct 22 10:30 'Program Files'/
drwxr-xr-x 1 BBM 197121          0 Oct 20 12:50 'Program Files (x86)'/
drwxr-xr-x 1 BBM 197121          0 Oct 20 13:17 ProgramData/
drwxr-xr-x 1 BBM 197121          0 Oct 22 13:05 prueba_git/
drwxr-xr-x 1 BBM 197121          0 Oct 20 12:19 Recovery/
-rw-r--r-- 1 BBM 197121 268435456 Oct 20 20:29 swapfile.sys
drwxr-xr-x 1 BBM 197121          0 Oct 20 11:22 'System Volume
Information'/
drwxr-xr-x 1 BBM 197121          0 Oct 20 12:32 Users/
drwxr-xr-x 1 BBM 197121          0 Oct 22 10:28 Windows/

BBM@DESKTOP-7V7495B MINGW64 /c
$ cd ./Default_databases/

BBM@DESKTOP-7V7495B MINGW64 /c/Default_databases
$ ls -l
total 2196
-rw-r--r-- 1 BBM 197121 2115268 Oct 20 20:33 InstNwnd.sql
-rw-r--r-- 1 BBM 197121 128520 Oct 20 20:33 InstPubs.sql

BBM@DESKTOP-7V7495B MINGW64 /c/Default_databases /* COPIAMOS ARCHIVO EN
DIRECTORIO NUEVO */
$ cp InstPubs.sql ../prueba_git

BBM@DESKTOP-7V7495B MINGW64 /c/prueba_git /* DICE QUE EN EL REPOSITORIO
HAY UN ARCHIVO TRACKEADO PERO NO COMMIT (SNAPSHOTTEADO) */
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    InstPubs.sql

nothing added to commit but untracked files present (use "git add" to
track)

BBM@DESKTOP-7V7495B MINGW64 /c/prueba_git
$

BBM@DESKTOP-7V7495B MINGW64 /c/prueba_git /* AGREGAS EL FICHERO A LA
ZONA INDEX */
$ git add .

BBM@DESKTOP-7V7495B MINGW64 /c/prueba_git

```

```
$ git status
On branch master

No commits yet

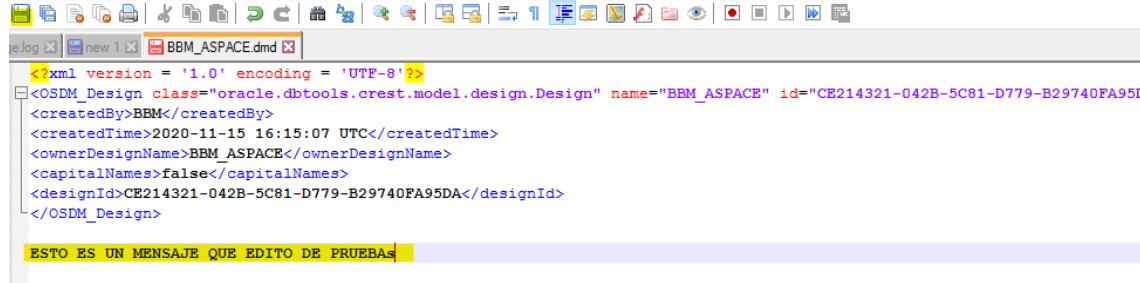
Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   InstPubs.sql

BBM@DESKTOP-7V7495B MINGW64 /c/prueba_git
$ git commit -m "Our first commit"
[master (root-commit) 7e0c175] Our first commit
  1 file changed, 2163 insertions(+)
  create mode 100644 InstPubs.sql

/* ENTRO EN EL NOTE PAD, DENTRO DE INSTPUBS Y LE CAMBIO EL NOMBRE. LO QUE
VAMOS A HACER ES DESHACER LOS CAMBIOS "PORQUE NO ME HA GUSTADO LO HECHO"
*/
```

```
BBM@DESKTOP-7V7495B MINGW64 /c/prueba_git /* PARA QUE SE VEA LA
MODIFICACIÓN*/
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   InstPubs.sql

no changes added to commit (use "git add" and/or "git commit -a")
```



```
BBM@DESKTOP-7V7495B MINGW64 /c/prueba_git /* LO USAMOS PARA DESHACER
LOS CAMBIOS */
$ git checkout InstPubs.sql
Updated 1 path from the index
```

```
BBM@DESKTOP-7V7495B MINGW64 /c/prueba_git /* COMPROBACIÓN Y VEMOS QUE EL
FICHERO ESTÁ COMO ANTES DEL CAMBIO*/
$ git status
On branch master
nothing to commit, working tree clean
```

```
<?xml version = '1.0' encoding = 'UTF-8'?>
<OSDM_Design class="oracle.dbtools.crest.model.design.Design" name="BBM_ASPACE" id="CE214321-042B-5C81-D779-B29740FA95DA">
<createdBy>BBM</createdBy>
<createdTime>2020-11-15 16:15:07 UTC</createdTime>
<ownerDesignName>BBM_ASPACE</ownerDesignName>
<capitalNames>false</capitalNames>
<designId>CE214321-042B-5C81-D779-B29740FA95DA</designId>
</OSDM_Design>
```

ESTO ES UN MENSAJE QUE EDITO DE PRUEBAS

```
<?xml version = '1.0' encoding = 'UTF-8'?>
<OSDM_Design class="oracle.dbtools.crest.model.design.Design" name="BBM_ASPACE" id="CE214321-042B-5C81-D779-B29740FA95DA">
<createdBy>BBM</createdBy>
<createdTime>2020-11-15 16:15:07 UTC</createdTime>
<ownerDesignName>BBM_ASPACE</ownerDesignName>
<capitalNames>false</capitalNames>
<designId>CE214321-042B-5C81-D779-B29740FA95DA</designId>
</OSDM_Design>
```

```
BBM@DESKTOP-7V7495B MINGW64 /c/prueba_git
$ git log
commit 7e0c17537eb71b9ded1eeb1239c0f518bedc5e19 (HEAD -> master) /*
COMMIT --> CREA UNOS CÓDIGOS (SHAR) QUE HACEN QUE ESE COMMIT SEAN ÚNICOS.
CON LA INFORMACIÓN QUE EXPIDE SE PUEDE IDENTIFICAR QUIÉN HIZO LAS
OPERACIONES */
Author: beadebmanuel <clase.bbm@gmail.com>
Date:   Thu Oct 22 17:01:52 2020 +0200
```

Our **first commit**

```
BBM@DESKTOP-7V7495B MINGW64 /c/prueba_git /* ME DICE TODAS LAS
OPERACIONES QUE FUÍ REALIZANDO HEAD ES COMO UN PUNTERO QUE APUNTA AL
COMMIT */
$ git reflog
7e0c175 (HEAD -> master) HEAD@{0}: commit (initial): Our first commit
```

```
/* AHORA HACEMOS LO MISMO CON InstNwnd.sql*/
```

```
BBM@DESKTOP-7V7495B MINGW64 /c/prueba_git
$ cd

BBM@DESKTOP-7V7495B MINGW64 ~
$ cd /c

BBM@DESKTOP-7V7495B MINGW64 /c
$ cd ./Default_databases/

BBM@DESKTOP-7V7495B MINGW64 /c/Default_databases
$ cp InstNwnd.sql ../prueba_git/
```

```
BBM@DESKTOP-7V7495B MINGW64 /c/Default_databases
$ cd
```

```
BBM@DESKTOP-7V7495B MINGW64 ~
$ cd /c
```

```
BBM@DESKTOP-7V7495B MINGW64 /c
$ cd ./prueba_git/

BBM@DESKTOP-7V7495B MINGW64 /c/prueba_git
$ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    InstNwnd.sql

nothing added to commit but untracked files present (use "git add" to
track)

BBM@DESKTOP-7V7495B MINGW64 /c/prueba_git
$ git add InstNwnd.sql

BBM@DESKTOP-7V7495B MINGW64 /c/prueba_git
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   InstNwnd.sql

BBM@DESKTOP-7V7495B MINGW64 /c/prueba_git
$ git commit -m "Our second commit"
[master 2e73812] Our second commit
  1 file changed, 0 insertions(+), 0 deletions(-)
  create mode 100644 InstNwend.sql

BBM@DESKTOP-7V7495B MINGW64 /c/prueba_git
$ git log
commit 2e7381298419405431d1b082041bd9f278a6b107 (HEAD -> master)
Author: beadebmanuel <clase.bbm@gmail.com>
Date:   Thu Oct 22 17:28:08 2020 +0200

Our second commit

commit 7e0c17537eb71b9ded1eeb1239c0f518bedc5e19
Author: beadebmanuel <clase.bbm@gmail.com>
Date:   Thu Oct 22 17:01:52 2020 +0200

Our first commit

BBM@DESKTOP-7V7495B MINGW64 /c/prueba_git
$ git reflog
2e73812 (HEAD -> master) HEAD@{0}: commit: Our second commit
7e0c175 HEAD@{1}: commit (initial): Our first commit

/* AHORA LO TENGO EN LOCAL, PERO QUIERO SUBIRLO A MI REPOSITORIO (EN
GITHUB, GITLAB O GITHUB). LA SENTENCIA PARA MANDAR LO QUE TENGO EN EL
REPOSITORIO ACTUAL AL REPOSITORIO REMOTO */

BBM@DESKTOP-7V7495B MINGW64 /c/prueba_git
$ git remote add origin prueba_git
```

```
BBM@DESKTOP-7V7495B MINGW64 /c/prueba_git /* AGREGAR UN REMOTO NUEVO*/
$ git remote add origin https://github.com/BBMASAX/BBM.git
fatal: remote origin already exists.
```

```
BBM@DESKTOP-7V7495B MINGW64 /c/prueba_git /* PARA COMPROBAR CUAL ES EL
ORIGIN Y HACIA DONDE APUNTA EL HEAD */
$ git remote show origin
* remote origin
  Fetch URL: https://github.com/BBMASAX/BBM.git
  Push URL: https://github.com/BBMASAX/BBM.git
  HEAD branch: master
  Remote branch:
    master new (next fetch will store in remotes/origin)
  Local ref configured for 'git push':
    master pushes to master (local out of date)
```

```
BBM@DESKTOP-7V7495B MINGW64 /c/prueba_git /* VERIFICAR EL NUEVO REMOTO */
$ git remote -v
origin https://github.com/BBMASAX/BBM.git (fetch)
origin https://github.com/BBMASAX/BBM.git (push)

/* FALLO COMENTADO */
```

```
BBM@DESKTOP-7V7495B MINGW64 /c/prueba_git /*NO ME DEJA SUBIR EL ARCHIVO
PORQUE YA HAY UN TRABAJO DENTRO HECHO POR MÍ (NO TENGO CLARO PORQUÉ)*/
$ git push -u origin master
To https://github.com/BBMASAX/BBM.git
 ! [rejected]           master -> master (fetch first)
error: failed to push some refs to 'https://github.com/BBMASAX/BBM.git'
hint: Updates were rejected because the remote contains work that you do
hint: not have locally. This is usually caused by another repository
pushing
hint: to the same ref. You may want to first integrate the remote changes
hint: (e.g., 'git pull ...') before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for
details.
```

```
BBM@DESKTOP-7V7495B MINGW64 /c/prueba_git /* CON ESTE COMANDO
"SOBREESCRIBÍ" LOS CAMBIOS REMOTOS DIRECTAMENTE CON LO QUE HAY EN MI
REPOSITORIO LOCAL*/
$ git push -f http://github.com/BBMASAX/BBM.git
warning: redirecting to https://github.com/BBMASAX/BBM.git/
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 2 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (6/6), 278.18 KiB | 776.00 KiB/s, done.
Total 6 (delta 0), reused 0 (delta 0), pack-reused 0
To http://github.com/BBMASAX/BBM.git
 + cacb0ba...22a3ac4 master -> master (forced update)
```

```
BBM@DESKTOP-7V7495B MINGW64 /c/prueba_git
$ git push -u origin master
```

Everything up-to-date
Branch 'master' set up to track remote branch 'master' from 'origin'.

/*HACEMOS LA PUEBA DE GIT CLONE*/

BBM@DESKTOP-7V7495B MINGW64 ~
\$ cd /c

BBM@DESKTOP-7V7495B MINGW64 /c
\$ ls
'\$Recycle.Bin'|/
'\$WINDOWS.~BT'|/
'\$WINRE_BACKUP_PARTITION.MARKER'|/
BBM_Repository_clone/
bootmgr
BOOTNXT
Default_databases/
'Documents and Settings'|@
Ficheros_resetrevert/
pagefile.sys
PerfLogs/

'Program Files'|/
'Program Files (x86)'|/
ProgramData/
prueba_git/
Recovery/
rstrvt_git/
swapfile.sys
'System Volume Information'|/
Users/
Windows/

BBM@DESKTOP-7V7495B MINGW64 /c
\$ cd ./BBM_Repository_clone/

BBM@DESKTOP-7V7495B MINGW64 /c/BBM_Repository_clone
\$ git clone https://github.com/BBMASAX/BBM.git
Cloning into 'BBM'...
remote: Enumerating objects: 6, done.
remote: Counting objects: 100% (6/6), done.
remote: Compressing objects: 100% (5/5), done.
remote: Total 6 (delta 0), reused 6 (delta 0), pack-reused 0
Receiving objects: 100% (6/6), 278.18 KiB | 1.24 MiB/s, done.

Creación del archivo README.md

Un archivo **README**, junto con una licencia de repositorio, pautas de contribución y un código de conducta, te ayudan a comunicar las expectativas y a administrar las contribuciones para tu proyecto.

Un archivo *README* suele ser el primer elemento que verá un visitante cuando entre a tu repositorio. Los archivos README habitualmente incluyen información sobre:

- Qué hace el proyecto.
- Por qué el proyecto es útil.
- Cómo pueden comenzar los usuarios con el proyecto.
- Dónde pueden recibir ayuda los usuarios con tu proyecto
- Quién mantiene y contribuye con el proyecto.

Si colocas tu archivo *README* en la raíz de tu repositorio, *docs*, o en el directorio oculto *.github*, GitHub lo reconocerá y automáticamente expondrá tu archivo *README* a los visitantes del repositorio

```
/* CREAR ARCHIVO README.md CON GIT BASH*/
BBM@DESKTOP-7V7495B MINGW64 /c/BBM_Repository_clone
$ cd ..

BBM@DESKTOP-7V7495B MINGW64 /c
$ cd ./prueba_git

BBM@DESKTOP-7V7495B MINGW64 /c/prueba_git (master)
$ touch README.md

BBM@DESKTOP-7V7495B MINGW64 /c/prueba_git (master)
$ git add README.md

BBM@DESKTOP-7V7495B MINGW64 /c/prueba_git (master)
$ git status
On branch master
Your branch is up to date with 'origin/master'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   README.md

BBM@DESKTOP-7V7495B MINGW64 /c/prueba_git (master)
$ git commit -m "Initial commit"
[master 0d127ec] Initial commit
  1 file changed, 0 insertions(+), 0 deletions(-)
  create mode 100644 README.md

BBM@DESKTOP-7V7495B MINGW64 /c/prueba_git (master)
$ git push -u origin master
```

```

Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 2 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 311 bytes | 311.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/BBMASAX/BBM.git
  22a3ac4..0d127ec master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'.

```

Ignorar archivos con .gitignore

Un archivo de tipo `.gitignore` va colocado en el directorio raíz del repositorio para indicarle a Git qué archivos y directorios ignorar cuando se hace una confirmación. Para compartir las reglas para ignorar con otros usuarios que clonian el repositorio, hay que confirmar el archivo de tipo `.gitignore` en el repositorio.

```

/* PRIMERO CREAMOS UN ARCHIVO .gitignore EN EL NOTE PAD*/
start notepad .gitignore

Dentro le metemos lo siguiente → *bak
/*Guardamos y lo subimos. Con esto, el sistema trackea lo que hay dentro
del archivo y lo ignora*/

BBM@DESKTOP-7V7495B MINGW64 /c/prueba_git (master)
$ git status
On branch master
Your branch is up to date with 'origin/master'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    .gitignore.txt

nothing added to commit but untracked files present (use "git add" to
track)

BBM@DESKTOP-7V7495B MINGW64 /c/prueba_git (master)
$ git add .gitignore.txt

BBM@DESKTOP-7V7495B MINGW64 /c/prueba_git (master)
$ git commit -m "upload git ignore"
[master 329e244] upload git ignore
 1 file changed, 1 insertion(+)
 create mode 100644 .gitignore.txt

```

Reset vs Revert

Cuando se mantiene el código usando sistemas de control de versiones como **git**, es inevitable que necesitemos deshacer algunos *commit* debido a errores o revertir el código temporal. En este caso hay diferentes formas de hacerlo, de las cuales mostraré dos:

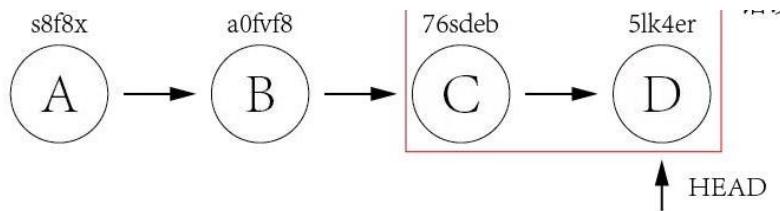
- git reset
- git revert

}

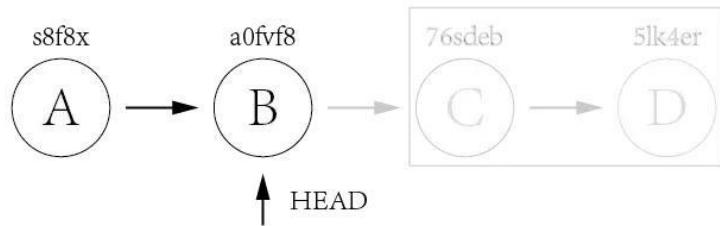
¿Cuál es la diferencia?

GIT RESET

Commit A y *B* son correctos, pero los *commit C* and *D* están mal. Ahora quiero volver al *commit B* y desechar *C* y *D*. Ahora mismo **HEAD** apunta al *commit D 5lk4er*, pero necesito que apunte al *commit B a0fvf8*. Para lograr este objetivo utilizo el comando **git reset -hard a0fvf8**.



Una vez ejecutado, el puntero **HEAD** vuelve a estar en *B*



Pero ahora el origen remoto todavía tiene el puntero **HEAD** para confirmar *D*, si usamos directamente **git push**, no actualizará el repositorio remoto, necesitamos agregar una opción **-f** para forzar la inserción de los cambios: **git push -f**. El inconveniente de este método es que todas las confirmaciones posteriores a **HEAD** desaparecerán una vez que se realice el reinicio. En caso de que un día descubramos que algunos de los commits se comieron buenos y queramos conservarlos, ya es demasiado tarde.

/*AHORA HAGO UNA PRUEBA PROPIA CON GIT RESET Y GIT REVERT*/

```
BBM@DESKTOP-7V7495B MINGW64 ~ /* CREO MI DIRECTORIO */
$ cd /c
```

```
BBM@DESKTOP-7V7495B MINGW64 /c
$ mkdir rstrvt_git

BBM@DESKTOP-7V7495B MINGW64 /c
$ cd ./rstrvt_git/

BBM@DESKTOP-7V7495B MINGW64 /c/rstrvt_git
$ git init
Initialized empty Git repository in C:/rstrvt_git/.git/

/*CREO UN FICHERO DE TEXTO DE PRUEBA*/
BBM@DESKTOP-7V7495B MINGW64 /c/rstrvt_git
$ start notepad fichero1 /* SE ABRE INMEDIATAMENTE UN NOTEBOOK PARA EDITAR
EL ARCHIVO*/

BBM@DESKTOP-7V7495B MINGW64 /c/rstrvt_git
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    fichero1.txt

nothing added to commit but untracked files present (use "git add" to
track)

BBM@DESKTOP-7V7495B MINGW64 /c/rstrvt_git
$ git add .

BBM@DESKTOP-7V7495B MINGW64 /c/rstrvt_git /* PARA DEPURAR LA ESCRITURA */
$ clear

$BBM@DESKTOP-7V7495B MINGW64 /c/rstrvt_git
$ git commit -m "Primer Fichero 1"
[master (root-commit) 07ec033] Primer Fichero 1
 1 file changed, 1 insertion(+)
 create mode 100644 fichero1.txt

BBM@DESKTOP-7V7495B MINGW64 /c/rstrvt_git /* SACA LOS LOG DE LINEA EN
LINEA */
$ git log --oneline
07ec033 (HEAD -> master) Primer Fichero 1

BBM@DESKTOP-7V7495B MINGW64 /c/rstrvt_git
$ git reflog
07ec033 (HEAD -> master) HEAD@{0}: commit (initial): Primer Fichero 1

/* AÑADO UN SEGUNDO FICHERO EN EL DIRECTORIO DE TRABAJO*/
BBM@DESKTOP-7V7495B MINGW64 /c/rstrvt_git
$ start notepad fichero2

BBM@DESKTOP-7V7495B MINGW64 /c/rstrvt_git
```

```
$ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    fichero2.txt

nothing added to commit but untracked files present (use "git add" to
track)

BBM@DESKTOP-7V7495B MINGW64 /c/rstrvt_git
$ git add .

BBM@DESKTOP-7V7495B MINGW64 /c/rstrvt_git
$ git commit -m "Segundo Fichero 2"
[master afa29c0] Segundo Fichero 2
  1 file changed, 2 insertions(+)
 create mode 100644 fichero2.txt

BBM@DESKTOP-7V7495B MINGW64 /c/rstrvt_git
$ git log --oneline
afa29c0 (HEAD -> master) Segundo Fichero 2
07ec033 Primer Fichero 1

BBM@DESKTOP-7V7495B MINGW64 /c/rstrvt_git
$ git reflog
afa29c0 (HEAD -> master) HEAD@{0}: commit: Segundo Fichero 2
07ec033 HEAD@{1}: commit (initial): Primer Fichero 1

/* CREAMOS EL FICHERO 3 Y LE APLICAMOS UN GIT RESET HARD PARA QUE SE VEA
LO QUE HACE*/
BBM@DESKTOP-7V7495B MINGW64 /c/rstrvt_git
$ start notepad fichero3

BBM@DESKTOP-7V7495B MINGW64 /c/rstrvt_git
$ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    fichero3.txt

nothing added to commit but untracked files present (use "git add" to
track)

BBM@DESKTOP-7V7495B MINGW64 /c/rstrvt_git
$ git add .

BBM@DESKTOP-7V7495B MINGW64 /c/rstrvt_git
$ git commit -m "Tercer Fichero 3"
[master efc9e0f] Tercer Fichero 3
  1 file changed, 1 insertion(+)
 create mode 100644 fichero3.txt

BBM@DESKTOP-7V7495B MINGW64 /c/rstrvt_git
$ git log --oneline
efc9e0f (HEAD -> master) Tercer Fichero 3
afa29c0 Segundo Fichero 2
07ec033 Primer Fichero 1
```

```
BBM@DESKTOP-7V7495B MINGW64 /c/rstrvt_git
$ git reflog
efc9e0f (HEAD -> master) HEAD@{0}: commit: Tercer Fichero 3
afa29c0 HEAD@{1}: commit: Segundo Fichero 2
07ec033 HEAD@{2}: commit (initial): Primer Fichero 1

BBM@DESKTOP-7V7495B MINGW64 /c/rstrvt_git /*EL HISTORIAL DE
CONFIRMACIONES SE RESTABLECE A ESTA CONFIRMACIÓN ESPECÍFICA*/
$ git reset --hard
HEAD is now at efc9e0f Tercer Fichero 3

BBM@DESKTOP-7V7495B MINGW64 /c/rstrvt_git
$ git log --oneline
efc9e0f (HEAD -> master) Tercer Fichero 3
afa29c0 Segundo Fichero 2
07ec033 Primer Fichero 1

BBM@DESKTOP-7V7495B MINGW64 /c/rstrvt_git /* PARA QUE SE VEA BIEN, LO
HAGO HACIA EL PRIMER FICHERO*/
$ git reset --hard 07ec033
HEAD is now at 07ec033 Primer Fichero 1

BBM@DESKTOP-7V7495B MINGW64 /c/rstrvt_git
$ git log --oneline
07ec033 (HEAD -> master) Primer Fichero 1

BBM@DESKTOP-7V7495B MINGW64 /c/rstrvt_git
$ git reflog
07ec033 (HEAD -> master) HEAD@{0}: reset: moving to 07ec033
efc9e0f HEAD@{1}: reset: moving to HEAD
efc9e0f HEAD@{2}: commit: Tercer Fichero 3
afa29c0 HEAD@{3}: commit: Segundo Fichero 2
07ec033 (HEAD -> master) HEAD@{4}: commit (initial): Primer Fichero 1

BBM@DESKTOP-7V7495B MINGW64 /c/rstrvt_git /* COMO SE PUEDE VER, AHORA
SOLO QUEDA EL PRIMER FICHERO */
$ ls
ficherol.txt

/*VUELVO A CREAR LOS FICHEROS DE TEXTO*/

BBM@DESKTOP-7V7495B MINGW64 /c/rstrvt_git
$ start notepad fichero2

BBM@DESKTOP-7V7495B MINGW64 /c/rstrvt_git
$ start notepad fichero3

BBM@DESKTOP-7V7495B MINGW64 /c/rstrvt_git
$ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    fichero2.txt
    fichero3.txt
```

```
nothing added to commit but untracked files present (use "git add" to track)

BBM@DESKTOP-7V7495B MINGW64 /c/rstrvt_git
$ git add fichero2.txt

BBM@DESKTOP-7V7495B MINGW64 /c/rstrvt_git
$ git commit -m "Segundo fichero 2"
[master 13fa448] Segundo fichero 2
 1 file changed, 1 insertion(+)
 create mode 100644 fichero2.txt

BBM@DESKTOP-7V7495B MINGW64 /c/rstrvt_git
$ git add fichero3.txt

BBM@DESKTOP-7V7495B MINGW64 /c/rstrvt_git
$ git commit -m "Tercer fichero 3"
[master 12e512a] Tercer fichero 3
 1 file changed, 1 insertion(+)
 create mode 100644 fichero3.txt

BBM@DESKTOP-7V7495B MINGW64 /c/rstrvt_git
$ git log --oneline
12e512a (HEAD -> master) Tercer fichero 3
13fa448 Segundo fichero 2
07ec033 Primer Fichero 1

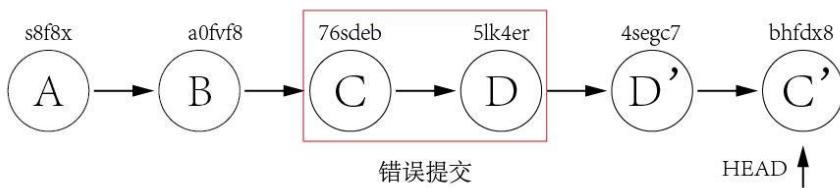
BBM@DESKTOP-7V7495B MINGW64 /c/rstrvt_git
$ git reflog
12e512a (HEAD -> master) HEAD@{0}: commit: Tercer fichero 3
13fa448 HEAD@{1}: commit: Segundo fichero 2
07ec033 HEAD@{2}: reset: moving to 07ec033
efc9e0f HEAD@{3}: reset: moving to HEAD
efc9e0f HEAD@{4}: commit: Tercer Fichero 3
afa29c0 HEAD@{5}: commit: Segundo Fichero 2
07ec033 HEAD@{6}: commit (initial): Primer Fichero 1

/*UNA VEZ PROBADO EL GIT RESET HARD EMPEZAMOS A PROBAR EL REVERT*/
```

GIT REVERT

El uso de `git revert` es crear una nueva confirmación que revierte una confirmación anterior. El HEAD apuntará a la nueva confirmación de reversión.

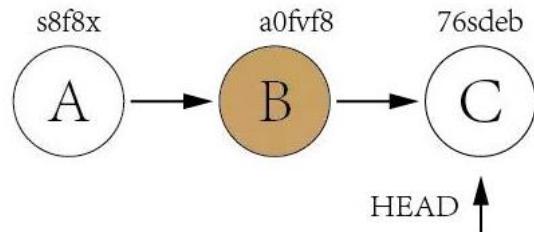
Para el ejemplo de git reset anterior, lo que tenemos que hacer es revertir commit D ([5lk4er](#)) y luego revertir el commit C ([76sdeb](#)). En ambos casos el comando **git revert número_commit**.



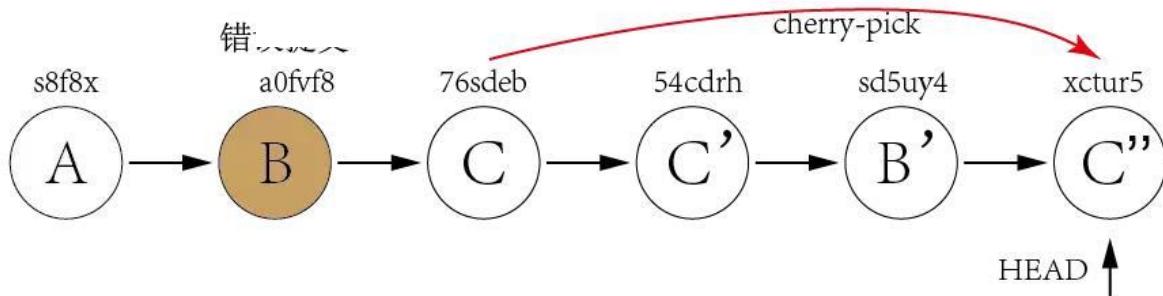
Con esto creamos dos nuevos commit, D' y C', cuyo puntero **HEAD** está apuntando a **bhfdx8**. En caso de querer hacer un rango utilizaríamos `git revert viejo_commit^..nuevo_commit`.

Este método no tendría la desventaja de `git reset`, apuntaría **HEAD** a la confirmación revertida recién creada y está bien enviar directamente los cambios a remoto sin usar la opción `-f`.

Para complicarlo un poco, asumamos que el *commit* malo es el B de ABC.



No es una buena idea usar `git reset` para revertir el *commit* B, ya que necesitamos mantener *commit* C. Ahora podemos revertir el *commit* C y B y luego usar *cherry-pick* para confirmar C nuevamente.



De la explicación anterior, podemos descubrir que la mayor diferencia entre `git reset` y `git revert` es que `git reset` restablecerá el estado de la rama a un estado anterior al eliminar todos los cambios posteriores a la confirmación deseada, mientras que `git revert` se restablecerá a un estado anterior. Estado creando nuevas confirmaciones reversibles y manteniendo las confirmaciones originales.

```

BBM@DESKTOP-7V7495B MINGW64 /c/rstrvt_git /* INTENTO BORRAR EL PRIMER
SEGUNDO FICHERO DE TEXTO QUE HICE Y NO ME DEJA*/

$ git revert fa29c0
fatal: bad revision 'fa29c0'

BBM@DESKTOP-7V7495B MINGW64 /c/rstrvt_git
$ git revert 13fa448
Removing fichero2.txt
[master c6f0d72] Revert "Segundo fichero 2"
 1 file changed, 1 deletion(-)
 delete mode 100644 fichero2.txt
  
```

```
BBM@DESKTOP-7V7495B MINGW64 /c/rstrvt_git
$ git log --oneline
c6f0d72 (HEAD -> master) Revert "Segundo fichero 2" /*MI PUNTERO HEAD
APUNTA AHORA A ESTE, EL CUAL NO EXISTE*/
12e512a Tercer fichero 3
13fa448 Segundo fichero 2
07ec033 Primer Fichero 1

BBM@DESKTOP-7V7495B MINGW64 /c/rstrvt_git /* COMO SE PUEDE VER LO HA
HECHO DESAPARECER*/
$ ls
ficherol.txt  fichero3.txt

BBM@DESKTOP-7V7495B MINGW64 /c/rstrvt_git /*LA DIFERENCIA ES QUE SI YO
HAGO REVERT OTRA VEZ, */
$ git revert c6f0d72
[master 5fc9162] Revert "Revert "Segundo fichero 2"""
 1 file changed, 1 insertion(+)
 create mode 100644 fichero2.txt

BBM@DESKTOP-7V7495B MINGW64 /c/rstrvt_git
$ ls
ficherol.txt  fichero2.txt  fichero3.txt

BBM@DESKTOP-7V7495B MINGW64 /c/rstrvt_git /* COMO SE VE, EN LOS LOG
QUEDAN REGISTRADOS*/
$ git log --oneline
5fc9162 (HEAD -> master) Revert "Revert "Segundo fichero 2"""
c6f0d72 Revert "Segundo fichero 2"
12e512a Tercer fichero 3
13fa448 Segundo fichero 2
07ec033 Primer Fichero 1

BBM@DESKTOP-7V7495B MINGW64 /c/rstrvt_git
$ git reflog
5fc9162 (HEAD -> master) HEAD@{0}: revert: Revert "Revert "Segundo
fichero 2"""
c6f0d72 HEAD@{1}: revert: Revert "Segundo fichero 2"
12e512a HEAD@{2}: commit: Tercer fichero 3
13fa448 HEAD@{3}: commit: Segundo fichero 2
07ec033 HEAD@{4}: reset: moving to 07ec033
efc9e0f HEAD@{5}: reset: moving to HEAD
efc9e0f HEAD@{6}: commit: Tercer Fichero 3
afa29c0 HEAD@{7}: commit: Segundo Fichero 2
07ec033 HEAD@{8}: commit (initial): Primer Fichero 1

BBM@DESKTOP-7V7495B MINGW64 /c/rstrvt_git /* UN APUNTE ES QUE SIEMPRE SE
ABRE UN ARCHIVO DE TEXTO */
$ git revert 07ec033
Removing ficherol.txt
[master 8be8a0b] Revert "Primer Fichero 1"
 1 file changed, 1 deletion(-)
 delete mode 100644 ficherol.txt

BBM@DESKTOP-7V7495B MINGW64 /c/rstrvt_git
```

```
$ ls
fichero2.txt  fichero3.txt

BBM@DESKTOP-7V7495B MINGW64 /c/rstrvt_git
$ git reflog
8be8a0b (HEAD -> master) HEAD@{0}: revert: Revert "Primer Fichero 1" /*
DONDE ESTÁ EL PUNTERO --> */
5fc9162 HEAD@{1}: revert: Revert "Segundo fichero 2"
c6f0d72 HEAD@{2}: revert: Revert "Segundo fichero 2"
12e512a HEAD@{3}: commit: Tercer fichero 3
13fa448 HEAD@{4}: commit: Segundo fichero 2
07ec033 HEAD@{5}: reset: moving to 07ec033
efc9e0f HEAD@{6}: reset: moving to HEAD
efc9e0f HEAD@{7}: commit: Tercer Fichero 3
afa29c0 HEAD@{8}: commit: Segundo Fichero 2
07ec033 HEAD@{9}: commit (initial): Primer Fichero 1

BBM@DESKTOP-7V7495B MINGW64 /c/rstrvt_git
$ git revert 8be8a0b
[master 10f2d62] Revert "Revert "Primer Fichero 1"""
 1 file changed, 1 insertion(+)
 create mode 100644 fichero1.txt

BBM@DESKTOP-7V7495B MINGW64 /c/rstrvt_git /* NUEVAMENTE HEMOS REVERTIDO
EL FICHERO*/
$ ls
fichero1.txt  fichero2.txt  fichero3.txt

BBM@DESKTOP-7V7495B MINGW64 /c/rstrvt_git
$ git revert HEAD /* 8be8a0b (HEAD -> master) HEAD@{0}: revert: Revert
"Primer Fichero 1" */
Removing fichero1.txt
[master 2d20f39] Revert "Revert "Revert "Primer Fichero 1"""
 1 file changed, 1 deletion(-)
 delete mode 100644 fichero1.txt

BBM@DESKTOP-7V7495B MINGW64 /c/rstrvt_git /* POR ESO QUEDAN ESTOS DOS
FICHEROS*/
$ ls
fichero2.txt  fichero3.txt

BBM@DESKTOP-7V7495B MINGW64 /c/rstrvt_git
$ git reflog
2d20f39 (HEAD -> master) HEAD@{0}: revert: Revert "Revert "Revert "Primer
Fichero 1"""
10f2d62 HEAD@{1}: revert: Revert "Primer Fichero 1"
8be8a0b HEAD@{2}: revert: Revert "Primer Fichero 1"
5fc9162 HEAD@{3}: revert: Revert "Segundo fichero 2"
c6f0d72 HEAD@{4}: revert: Revert "Segundo fichero 2"
12e512a HEAD@{5}: commit: Tercer fichero 3
13fa448 HEAD@{6}: commit: Segundo fichero 2
07ec033 HEAD@{7}: reset: moving to 07ec033
efc9e0f HEAD@{8}: reset: moving to HEAD
efc9e0f HEAD@{9}: commit: Tercer Fichero 3
afa29c0 HEAD@{10}: commit: Segundo Fichero 2
07ec033 HEAD@{11}: commit (initial): Primer Fichero 1
```

```
BBM@DESKTOP-7V7495B MINGW64 /c/rstrvt_git
$ git revert 2d20f39
[master 7c9b1d1] Revert "Revert "Revert "Revert "Revert "Primer Fichero 1"""
 1 file changed, 1 insertion(+)
 create mode 100644 fichero1.txt

BBM@DESKTOP-7V7495B MINGW64 /c/rstrvt_git
$ ls
fichero1.txt  fichero2.txt  fichero3.txt

BBM@DESKTOP-7V7495B MINGW64 /c/rstrvt_git
$ git log --oneline
7c9b1d1 (HEAD -> master) Revert "Revert "Revert "Revert "Revert "Primer Fichero
1"""
2d20f39 Revert "Revert "Revert "Primer Fichero 1"""
10f2d62 Revert "Revert "Primer Fichero 1"""
8be8a0b Revert "Primer Fichero 1"
5fc9162 Revert "Revert "Segundo fichero 2"""
c6f0d72 Revert "Segundo fichero 2"
12e512a Tercer fichero 3
13fa448 Segundo fichero 2
07ec033 Primer Fichero 1

/*RETOMAMOS LA PARTE DE RESET*/

BBM@DESKTOP-7V7495B MINGW64 /c/rstrvt_git /* LO DEVUELVO AL PRIMER
SEGUNDO TXT CREADO*/
$ git reset 13fa448

BBM@DESKTOP-7V7495B MINGW64 /c/rstrvt_git
$ git log --oneline
13fa448 (HEAD -> master) Segundo fichero 2
07ec033 Primer Fichero 1

BBM@DESKTOP-7V7495B MINGW64 /c/rstrvt_git
$ ls
fichero1.txt  fichero2.txt  fichero3.txt

BBM@DESKTOP-7V7495B MINGW64 /c/rstrvt_git
$ start notepad fichero2.txt

$BBM@DESKTOP-7V7495B MINGW64 /c/rstrvt_git
$ git reset --hard 07ec033
HEAD is now at 07ec033 Primer Fichero 1

BBM@DESKTOP-7V7495B MINGW64 /c/rstrvt_git
$ las
bash: las: command not found

BBM@DESKTOP-7V7495B MINGW64 /c/rstrvt_git
$ ls
fichero1.txt  fichero3.txt
```

```
BBM@DESKTOP-7V7495B MINGW64 /c/rstrvt_git
$ git log --oneline
07ec033 (HEAD -> master) Primer Fichero 1

BBM@DESKTOP-7V7495B MINGW64 /c/rstrvt_git
$ git reflog
07ec033 (HEAD -> master) HEAD@{0}: reset: moving to 07ec033
13fa448 HEAD@{1}: reset: moving to 13fa448
7c9b1d1 HEAD@{2}: revert: Revert "Revert "Revert "Revert "Primer Fichero 1"""
2d20f39 HEAD@{3}: revert: Revert "Revert "Revert "Primer Fichero 1"""
10f2d62 HEAD@{4}: revert: Revert "Revert "Primer Fichero 1"""
8be8a0b HEAD@{5}: revert: Revert "Primer Fichero 1"
5fc9162 HEAD@{6}: revert: Revert "Segundo fichero 2"""
c6f0d72 HEAD@{7}: revert: Revert "Segundo fichero 2"
12e512a HEAD@{8}: commit: Tercer fichero 3
13fa448 HEAD@{9}: commit: Segundo fichero 2
07ec033 (HEAD -> master) HEAD@{10}: reset: moving to 07ec033
efc9e0f HEAD@{11}: reset: moving to HEAD
efc9e0f HEAD@{12}: commit: Tercer Fichero 3
afa29c0 HEAD@{13}: commit: Segundo Fichero 2
07ec033 (HEAD -> master) HEAD@{14}: commit (initial): Primer Fichero 1

BBM@DESKTOP-7V7495B MINGW64 /c/rstrvt_git /*DEL PRIMER FICHERO*/
$ git reset --hard 07ec033
HEAD is now at 07ec033 Primer Fichero 1

BBM@DESKTOP-7V7495B MINGW64 /c/rstrvt_git
$ ls
fichero1.txt  fichero3.txt

BBM@DESKTOP-7V7495B MINGW64 /c/rstrvt_git
$ git log --oneline
07ec033 (HEAD -> master) Primer Fichero 1

BBM@DESKTOP-7V7495B MINGW64 /c/rstrvt_git
$ git reflog
07ec033 (HEAD -> master) HEAD@{0}: reset: moving to 07ec033
07ec033 (HEAD -> master) HEAD@{1}: reset: moving to 07ec033
13fa448 HEAD@{2}: reset: moving to 13fa448
7c9b1d1 HEAD@{3}: revert: Revert "Revert "Revert "Primer Fichero 1"""
2d20f39 HEAD@{4}: revert: Revert "Revert "Revert "Primer Fichero 1"""
10f2d62 HEAD@{5}: revert: Revert "Revert "Primer Fichero 1"""
8be8a0b HEAD@{6}: revert: Revert "Primer Fichero 1"
5fc9162 HEAD@{7}: revert: Revert "Segundo fichero 2"""
c6f0d72 HEAD@{8}: revert: Revert "Segundo fichero 2"
12e512a HEAD@{9}: commit: Tercer fichero 3
13fa448 HEAD@{10}: commit: Segundo fichero 2
07ec033 (HEAD -> master) HEAD@{11}: reset: moving to 07ec033
efc9e0f HEAD@{12}: reset: moving to HEAD
efc9e0f HEAD@{13}: commit: Tercer Fichero 3
afa29c0 HEAD@{14}: commit: Segundo Fichero 2
07ec033 (HEAD -> master) HEAD@{15}: commit (initial): Primer Fichero 1
```

```
BBM@DESKTOP-7V7495B MINGW64 /c/rstrvt_git /*AÑADIMOS LA INFORMACIÓN TODA
A UN REPOSITORIO NUEVO*/
$ git remote add origin https://github.com/BBMASAX/resetrevert.git

BBM@DESKTOP-7V7495B MINGW64 /c/rstrvt_git
$ git remote -v
origin  https://github.com/BBMASAX/resetrevert.git (fetch)
origin  https://github.com/BBMASAX/resetrevert.git (push)

BBM@DESKTOP-7V7495B MINGW64 /c/rstrvt_git /* CREAMOS UN COMMIT DE TODO
ESTO */
$ git add .

BBM@DESKTOP-7V7495B MINGW64 /c/rstrvt_git
$ git commit -m "Otro commit del Fichero 2"
[master 3ab9f1e] Otro commit del Fichero 2
 1 file changed, 1 insertion(+)
 create mode 100644 fichero3.txt

BBM@DESKTOP-7V7495B MINGW64 /c/rstrvt_git
$ git log --oneline
3ab9f1e (HEAD -> master) Otro commit del Fichero 2
07ec033 Primer Fichero 1

BBM@DESKTOP-7V7495B MINGW64 /c/rstrvt_git
$ git status
On branch master
nothing to commit, working tree clean

manuel@DESKTOP-VPL66KC MINGW64 /c/probandoresetrevert (master)
$ ls
ficherol.txt  fichero2.txt

manuel@DESKTOP-VPL66KC MINGW64 /c/probandoresetrevert (master)
$ git reset --hard 974eb4b
HEAD is now at 974eb4b Primer Fichero 1

manuel@DESKTOP-VPL66KC MINGW64 /c/probandoresetrevert (master)
$ ls
ficherol.txt

manuel@DESKTOP-VPL66KC MINGW64 /c/probandoresetrevert (master)
$ git log --oneline
974eb4b (HEAD -> master) Primer Fichero 1

BBM@DESKTOP-7V7495B MINGW64 /c/rstrvt_git
$ git reset --hard 07ec033
HEAD is now at 07ec033 Primer Fichero 1

BBM@DESKTOP-7V7495B MINGW64 /c/rstrvt_git
$ ls
ficherol.txt

BBM@DESKTOP-7V7495B MINGW64 /c/rstrvt_git
$ git status
On branch master
```

```
nothing to commit, working tree clean

BBM@DESKTOP-7V7495B MINGW64 /c/rstrvt_git
$ git log --oneline
07ec033 (HEAD -> master) Primer Fichero 1

BBM@DESKTOP-7V7495B MINGW64 /c/rstrvt_git
$ git reflog
07ec033 (HEAD -> master) HEAD@{0}: reset: moving to 07ec033
3ab9f1e HEAD@{1}: reset: moving to HEAD
3ab9f1e HEAD@{2}: reset: moving to HEAD
3ab9f1e HEAD@{3}: commit: Otro commit del Fichero 2
07ec033 (HEAD -> master) HEAD@{4}: reset: moving to 07ec033
07ec033 (HEAD -> master) HEAD@{5}: reset: moving to 07ec033
13fa448 HEAD@{6}: reset: moving to 13fa448
7c9b1d1 HEAD@{7}: revert: Revert "Revert "Revert "Revert "Primer Fichero 1"""
2d20f39 HEAD@{8}: revert: Revert "Revert "Revert "Primer Fichero 1"""
10f2d62 HEAD@{9}: revert: Revert "Revert "Primer Fichero 1"""
8be8a0b HEAD@{10}: revert: Revert "Primer Fichero 1"
5fc9162 HEAD@{11}: revert: Revert "Revert "Segundo fichero 2"""
c6f0d72 HEAD@{12}: revert: Revert "Segundo fichero 2"
12e512a HEAD@{13}: commit: Tercer fichero 3
13fa448 HEAD@{14}: commit: Segundo fichero 2
07ec033 (HEAD -> master) HEAD@{15}: reset: moving to 07ec033
efc9e0f HEAD@{16}: reset: moving to HEAD
efc9e0f HEAD@{17}: commit: Tercer Fichero 3
afa29c0 HEAD@{18}: commit: Segundo Fichero 2
07ec033 (HEAD -> master) HEAD@{19}: commit (initial): Primer Fichero 1

BBM@DESKTOP-7V7495B MINGW64 /c/rstrvt_git
$ git reset --soft 13fa448

BBM@DESKTOP-7V7495B MINGW64 /c/rstrvt_git
$ ls
ficherol.txt

BBM@DESKTOP-7V7495B MINGW64 /c/rstrvt_git
$ git log --oneline
13fa448 (HEAD -> master) Segundo fichero 2
07ec033 Primer Fichero 1
```

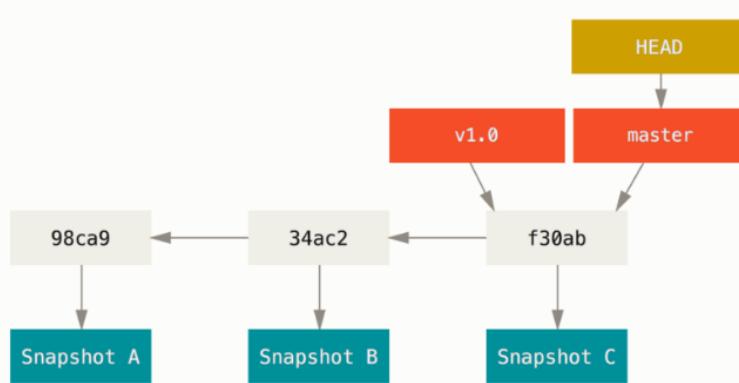
Git Branch

Cualquier sistema de control de versiones moderno tiene algún mecanismo para soportar el uso de **ramas**. Cuando hablamos de ramificaciones, significa que tú has tomado la rama principal de desarrollo (master) y a partir de ahí has continuado trabajando sin seguir la rama principal de desarrollo.

Para entender realmente cómo ramifica Git, previamente hemos de examinar la forma en que almacena sus datos.

En cada confirmación de cambios (commit), Git almacena una instantánea de tu trabajo preparado. Dicha instantánea contiene además unos metadatos con el autor y el mensaje explicativo, y uno o varios apuntadores a las confirmaciones (commit) que sean padres directos de esta (un parent en los casos de confirmación normal, y múltiples padres en los casos de estar confirmando una fusión (merge) de dos o más ramas).

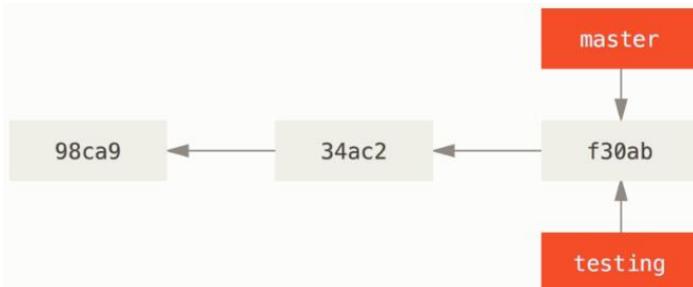
Una rama Git es simplemente un apuntador móvil apuntando a una de esas confirmaciones. La rama por defecto de Git es la rama master. Con la primera confirmación de cambios que realicemos, se creará esta rama principal master apuntando a dicha confirmación. En cada confirmación de cambios que realicemos, la rama irá avanzando automáticamente.



¿Qué sucede cuando creas una nueva rama? Bueno..., simplemente se crea un nuevo apuntador para que lo puedas mover libremente. Por ejemplo, supongamos que quieres crear una rama nueva denominada "testing". Para ello, usarás el comando git branch:

```
$ git branch testing
```

Esto creará un nuevo apuntador apuntando a la misma confirmación donde estés actualmente.



```
BBM@DESKTOP-7V7495B MINGW64 /c /*CREO UN NUEVO DIRECTORIO PARA HACER LAS PRUEBAS DE BRANCHES*/
$ mkdir ./prueba_branches
```

```
BBM@DESKTOP-7V7495B MINGW64 /c
$ cd ./prueba_branches
```

```
BBM@DESKTOP-7V7495B MINGW64 /c/prueba_branches
$ git init
Initialized empty Git repository in C:/prueba_branches/.git/
```

```
BBM@DESKTOP-7V7495B MINGW64 /c/prueba_branches (master) /*AL TRATAR DE CREAR UNA RAMA DISTINTA, NO ME DEJA HASTA QUE NO HAYA REALIZADO MI PRIMER
```

```
COMMIT. NO HAY NADA PARA HACER UNA RAMA SI NO HAY NADA EN EL  
REPOSOITORIO*/  
$ git branch experimento  
fatal: Not a valid object name: 'master'.  
  
BBM@DESKTOP-7V7495B MINGW64 /c/prueba_branches (master)  
$ git status  
On branch master  
  
No commits yet  
  
nothing to commit (create/copy files and use "git add" to track)  
  
/*PARA ESTA PRUEBA VAMOS A UTILIZAR LOS SIGUIENTES ELEMENTOS -->  
REPOSITORIO https://github.com/BBMASAX/resetrevert.git Y COPIAR PUBS*/  
  
BBM@DESKTOP-7V7495B MINGW64 /c/prueba_branches (master)  
$ cd ..  
  
BBM@DESKTOP-7V7495B MINGW64 /c  
$ cd ./Default_databases/  
  
BBM@DESKTOP-7V7495B MINGW64 /c/Default_databases /*COPIAMOS PUBS*/  
$ cp InstPubs.sql ../prueba_branches/  
  
BBM@DESKTOP-7V7495B MINGW64 /c/Default_databases  
$ cd ..  
  
BBM@DESKTOP-7V7495B MINGW64 /c  
$ cd ./prueba_branches/  
  
BBM@DESKTOP-7V7495B MINGW64 /c/prueba_branches (master)  
$ git status  
On branch master  
  
No commits yet  
  
Untracked files:  
  (use "git add <file>..." to include in what will be committed)  
    InstPubs.sql  
  
nothing added to commit but untracked files present (use "git add" to  
track)  
  
BBM@DESKTOP-7V7495B MINGW64 /c/prueba_branches (master)  
$ git add .  
  
BBM@DESKTOP-7V7495B MINGW64 /c/prueba_branches (master) /*HACEMOS EL  
COMMITTE Y COMPROBAMOS*/  
$ git commit -m "Primer Commit"  
[master (root-commit) 341d2b2] Primer Commit  
  1 file changed, 2163 insertions(+)  
  create mode 100644 InstPubs.sql  
  
BBM@DESKTOP-7V7495B MINGW64 /c/prueba_branches (master)
```

```
$ git log --oneline
341d2b2 (HEAD -> master) Primer Commit

BBM@DESKTOP-7V7495B MINGW64 /c/prueba_branches (master)
$ git reflog
341d2b2 (HEAD -> master) HEAD@{0}: commit (initial): Primer Commit

/* MODIFICO EL FICHERO
COMENTO ESTA LÍNEA
-- execute sp_dboption 'pubs' , 'trunc. log on chkpt.' , 'true'
-- GO*/

BBM@DESKTOP-7V7495B MINGW64 /c/prueba_branches (master) /*VEMOS QUE AHORA
HAY UN ARCHIVO PARA HACER COMMIT DESPUES DEL CAMBIO*/
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   InstPubs.sql

no changes added to commit (use "git add" and/or "git commit -a")

BBM@DESKTOP-7V7495B MINGW64 /c/prueba_branches (master) /*ESTE COMANDO SE
USA PARA LLEVAR A CABO UNA FUNCIÓN PARA ESTABLECER LAS DIFERENCIAS EN LAS
FUENTES DE DATOS DE GIT*/
$ git diff
diff --git a/InstPubs.sql b/InstPubs.sql
index d4ale96..bee63df 100644
--- a/InstPubs.sql
+++ b/InstPubs.sql
@@ -53,8 +53,8 @@ if db_name() <> 'pubs'

GO

-execute sp_dboption 'pubs' , 'trunc. log on chkpt.' , 'true'
-Go
+--execute sp_dboption 'pubs' , 'trunc. log on chkpt.' , 'true'
+--Go

execute sp_addtype id      , 'varchar(11)' , 'NOT NULL'
execute sp_addtype tid     , 'varchar(6)'  , 'NOT NULL'

BBM@DESKTOP-7V7495B MINGW64 /c/prueba_branches (master)
$ git add .

BBM@DESKTOP-7V7495B MINGW64 /c/prueba_branches (master)
$ git commit -m "Segundo commit Commit"
[master 3ea8047] Segundo commit Commit
 1 file changed, 2 insertions(+), 2 deletions(-)

BBM@DESKTOP-7V7495B MINGW64 /c/prueba_branches (master)
$ git log --oneline
3ea8047 (HEAD -> master) Segundo commit Commit
341d2b2 Primer Commit
```

```
BBM@DESKTOP-7V7495B MINGW64 /c/prueba_branches (master)
$ git reflog
3ea8047 (HEAD -> master) HEAD@{0}: commit: Segundo commit Commit
341d2b2 HEAD@{1}: commit (initial): Primer Commit

BBM@DESKTOP-7V7495B MINGW64 /c/prueba_branches (master) /*AL HACERLO
ENTRE DOS COMMITS, VARÍAMOS COMO COSA INTERESANTE LOS CAMBIOS QUE HEMOS
HECHO, EN ESTE CASO EN EL TXT*/
$ git diff 341d2b2..3ea8047
diff --git a/InstPubs.sql b/InstPubs.sql
index d4a1e96..bee63df 100644
--- a/InstPubs.sql
+++ b/InstPubs.sql
@@ -53,8 +53,8 @@ if db_name() <> 'pubs'

GO

-execute sp_dboption 'pubs' , 'trunc. log on chkpt.' , 'true'
-Go
+--execute sp_dboption 'pubs' , 'trunc. log on chkpt.' , 'true'
+--GO

execute sp_addtype id      , 'varchar(11)' , 'NOT NULL'
execute sp_addtype tid     , 'varchar(6)'   , 'NOT NULL'

BBM@DESKTOP-7V7495B MINGW64 /c/prueba_branches (master) /* CON ESTE
COMANDO MIRAMOS EN QUÉ RAMA NOS ENCONTRAMOS*/
$ git branch
* master

BBM@DESKTOP-7V7495B MINGW64 /c/prueba_branches (master) /* PARA VER LAS
RAMAS LOCALES*/
$ git show-branch
[master] Segundo commit Commit

BBM@DESKTOP-7V7495B MINGW64 /c/prueba_branches (master) /*CREO RAMA*/
$ git branch experimentorama

BBM@DESKTOP-7V7495B MINGW64 /c/prueba_branches (master) /*CON EL * VEMOS
EN QUÉ RAMA ESTOY*/
$ git branch
  experimentorama
* master

BBM@DESKTOP-7V7495B MINGW64 /c/prueba_branches (otroexperimento) /*CON
ESTE COMANDO ME CAMBIO A LA RAMA QUE QUIERO*/
$ git checkout experimentorama
Switched to branch 'experimentorama'

BBM@DESKTOP-7V7495B MINGW64 /c/prueba_branches (experimentorama)
$ git branch
* experimentorama
  master
  otroexperimento
```

```
BBM@DESKTOP-7V7495B MINGW64 /c/prueba_branches (master) /*CON ESTE
COMANDO LA CREO Y ME CAMBIO A LA RAMA*/
$ git checkout -b otroexperimento
Switched to a new branch 'otroexperimento'

BBM@DESKTOP-7V7495B MINGW64 /c/prueba_branches (otroexperimento)
$ git branch
  experimentorama
  master
* otroexperimento

BBM@DESKTOP-7V7495B MINGW64 /c/prueba_branches (experimentorama) /*ME
VOLVI A CAMBIAR A OTRA RAMA Y VEMOS CON STATUS QUE NO HAY NINGÚN COMMIT
HECHO*/
$ git status
On branch experimentorama
nothing to commit, working tree clean

/*HAGO UNA COPIA DE Northwind*/

BBM@DESKTOP-7V7495B MINGW64 /c/prueba_branches (experimentorama)
$ cd ..

BBM@DESKTOP-7V7495B MINGW64 /c
$ cd ./Default_databases/

BBM@DESKTOP-7V7495B MINGW64 /c/Default_databases
$ cp InstNwnd.sql Inst
InstNwnd.sql  InstPubsql

BBM@DESKTOP-7V7495B MINGW64 /c/Default_databases
$ cp InstNwnd.sql ../prueba_branches/

BBM@DESKTOP-7V7495B MINGW64 /c/Default_databases
$ cd ..

BBM@DESKTOP-7V7495B MINGW64 /c
$ cd ./prueba_branches/

BBM@DESKTOP-7V7495B MINGW64 /c/prueba_branches (experimentorama) /*AL
ESTAR EN OTRA RAMA ESTAMOS COPIANDO EN ELLA, NO EN MASTER*/
$ git status
On branch experimentorama
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    InstNwnd.sql

nothing added to commit but untracked files present (use "git add" to
track)

BBM@DESKTOP-7V7495B MINGW64 /c/prueba_branches (experimentorama)
/*AÑADIMOS UN COMMIT SOBRE ESTA RAMA*/
$ git add .

BBM@DESKTOP-7V7495B MINGW64 /c/prueba_branches (experimentorama)
$ git commit -m "Primer Commit Rama Experimental"
[experimentorama a4b57ca] Primer Commit Rama Experimental
```

```
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 InstNwnd.sql

BBM@DESKTOP-7V7495B MINGW64 /c/prueba_branches (experimentorama) /*AL
VOLVER A LA RAMA MASTER VEMOS QUE LOS CAMBIOS DE COMMIT SOLO HAY EL QUE
HICIMOS AL PRINCIPIO Y QUE TENEMOS DOS PUNTEROS*/
$ git checkout master
Switched to branch 'master'

BBM@DESKTOP-7V7495B MINGW64 /c/prueba_branches (master)
$ git status
On branch master
nothing to commit, working tree clean

BBM@DESKTOP-7V7495B MINGW64 /c/prueba_branches (master)
$ git log --oneline
3ea8047 (HEAD -> master, otroexperimento) Segundo commit Commit
341d2b2 Primer Commit

/* AHORA VAMOS A FUSIONAR LAS RAMAS EN UNA SOLA*/

BBM@DESKTOP-7V7495B MINGW64 /c/prueba_branches (master)
$ git checkout master
Already on 'master'

BBM@DESKTOP-7V7495B MINGW64 /c/prueba_branches (master) /* CON MERGE
HACEMOS LA FUSIÓN*/
$ git merge experimentorama
Updating 3ea8047..a4b57ca
Fast-forward
  InstNwnd.sql | Bin 0 -> 2115268 bytes
  1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 InstNwnd.sql

BBM@DESKTOP-7V7495B MINGW64 /c/prueba_branches (master)
$ git log --oneline
a4b57ca (HEAD -> master, experimentorama) Primer Commit Rama Experimental
/*AHORA VEMOS EL COMMIT HECHO EN LA RAMA EXPERIMENTAL*/
3ea8047 (otroexperimento) Segundo commit Commit
341d2b2 Primer Commit

BBM@DESKTOP-7V7495B MINGW64 /c/prueba_branches (master)
$ git reflog
a4b57ca (HEAD -> master, experimentorama) HEAD@{0}: merge
experimentorama: Fast-forward
3ea8047 (otroexperimento) HEAD@{1}: checkout: moving from master to
master
3ea8047 (otroexperimento) HEAD@{2}: checkout: moving from experimentorama
to master
a4b57ca (HEAD -> master, experimentorama) HEAD@{3}: commit: Primer Commit
Rama Experimental
3ea8047 (otroexperimento) HEAD@{4}: checkout: moving from otroexperimento
to experimentorama
3ea8047 (otroexperimento) HEAD@{5}: checkout: moving from master to
otroexperimento
3ea8047 (otroexperimento) HEAD@{6}: commit: Segundo commit Commit
341d2b2 HEAD@{7}: commit (initial): Primer Commit
```

```
BBM@DESKTOP-7V7495B MINGW64 /c/prueba_branches (master)
$ git merge master -m 'Un mensaje del merge de master en el branch
experimentorama'
Already up to date.

BBM@DESKTOP-7V7495B MINGW64 /c/prueba_branches (master)
$ git remote add origin https://github.com/BBMASAX/resetrevert.git

BBM@DESKTOP-7V7495B MINGW64 /c/prueba_branches (master) /*CON ESTO
FUSIONAMOS HACIA REMOTO*/
$ git push -u origin experimentorama
Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Delta compression using up to 2 threads
Compressing objects: 100% (7/7), done.
Writing objects: 100% (9/9), 278.43 KiB | 1.71 MiB/s, done.
Total 9 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), done.
To https://github.com/BBMASAX/resetrevert.git
 * [new branch]      experimentorama -> experimentorama
Branch 'experimentorama' set up to track remote branch 'experimentorama'
from 'origin'.
```

/*POR ÚLTIMO BORRAMOS LAS RAMAS*/

```
/*
git branch -d rama_a_borrar --> Sin embargo, puede que esta acción no nos
funcione porque hayamos hecho cambios que no se hayan salvado en el
repositorio remoto, o no se hayan fusionado con otras ramas. En el caso
que queramos forzar el borrado de la rama, para eliminarla
independientemente de si se ha hecho el push o el merge, tendrás que usar
la opción "-D".
```

```
git branch -D rama_a_borrar --> Prestar especial atención a esta opción
"-D", ya que al eliminar de este modo pueden haber cambios que ya no se
puedan recuperar. Como puedes apreciar, es bastante fácil de confundir
con "-d", opción más segura, ya que no permite borrado de ramas en
situaciones donde se pueda perder código.
```

```
*/
```

```
BBM@DESKTOP-7V7495B MINGW64 /c/prueba_branches (master)
$ git branch -d otroexperimento
Deleted branch otroexperimento (was 3ea8047).
```

```
BBM@DESKTOP-7V7495B MINGW64 /c/prueba_branches (master)
$ git branch
  experimentorama
* master
```

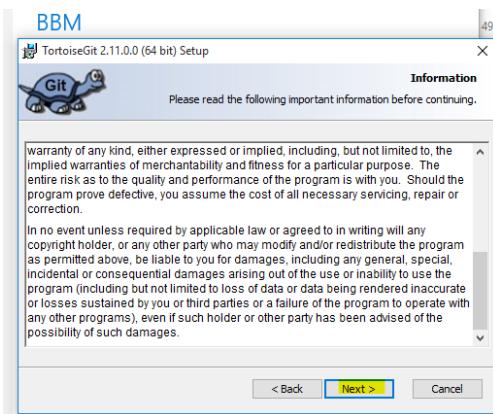
```
BBM@DESKTOP-7V7495B MINGW64 /c/prueba_branches (master)
```

Entorno Gráfico en Windows: GitTortoise

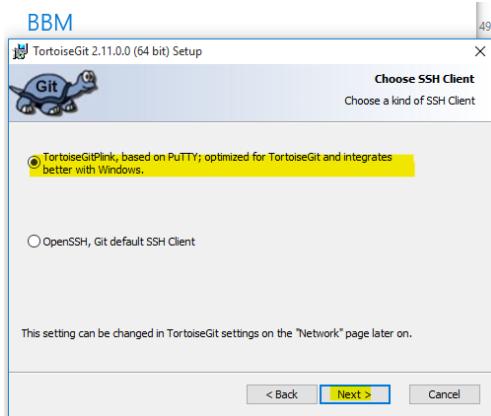
Comenzamos descargando la versión de 64 bits en la página <https://tortoisegit.org/download/>. Acto seguido le damos a instalar y nos aparece la primera pantalla, en la que haremos clic en el botón *Next*.



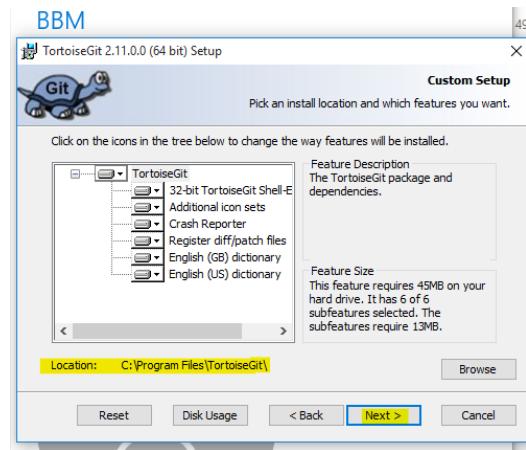
Leemos la información y clicamos en *Next*.



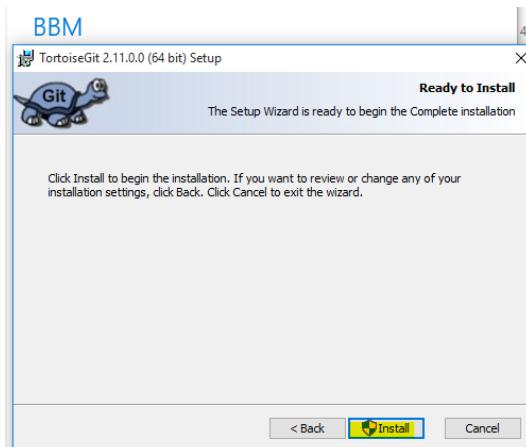
Ahora el resto de las ventanas es darle a *Next*. Realmente esta instalación no tiene mucha complicación, pero así también dejo muestra de este “entorno gráfico” para Windows.



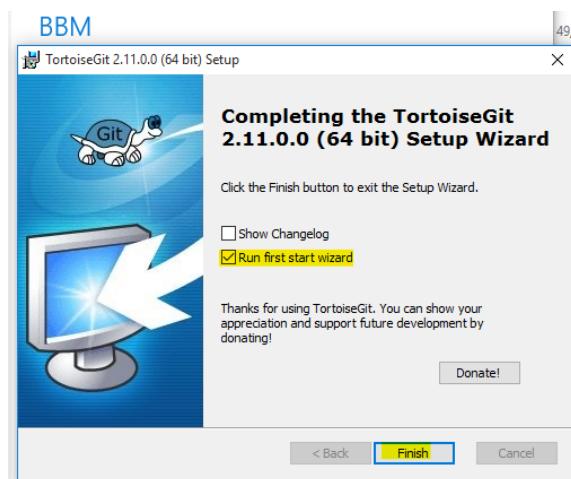
Dejamos por defecto la localización del Programa



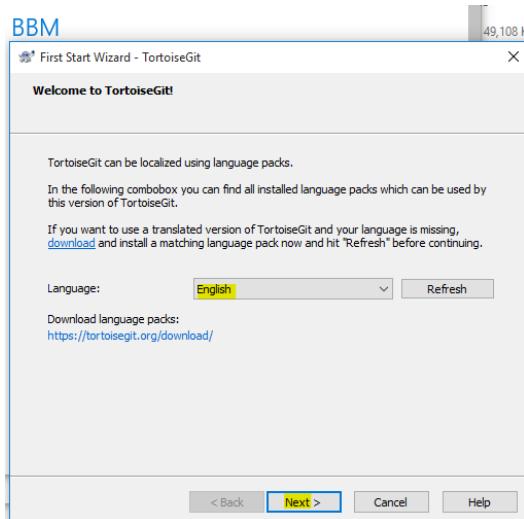
Hacemos click en *Install*. En este momento vamos a tener que aceptar la instalación con nuestros permisos de administrador



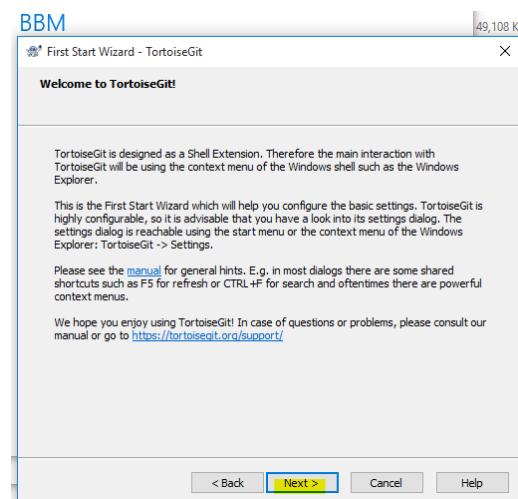
Por último le damos a finalizar y dejamos marcado el *check* conforme queremos que se inicie el asistente:



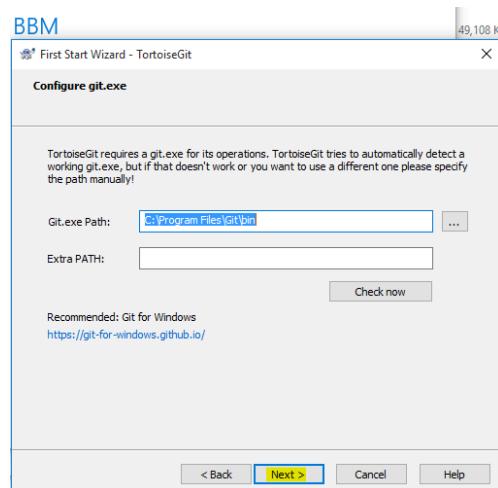
En este asistente, tendremos que elegir el idioma



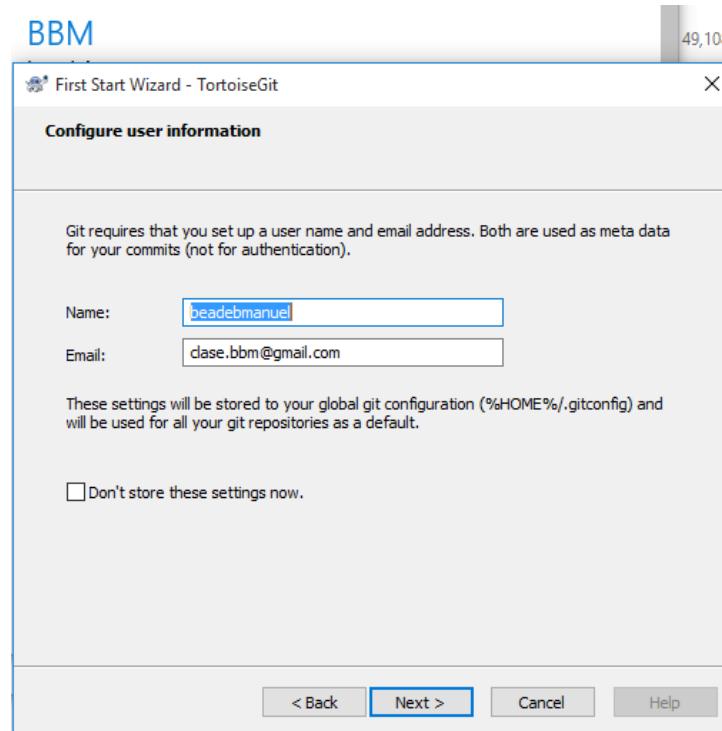
La siguiente página es una bienvenida a Tortoisegit:



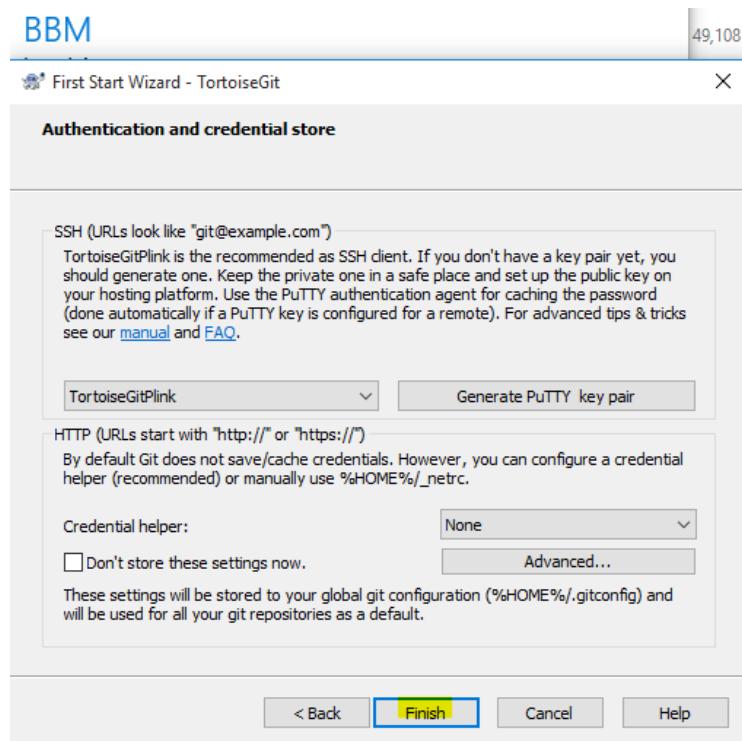
Elegimos dónde queremos meter el archivo **git.exe** (el cual dejamos por defecto):



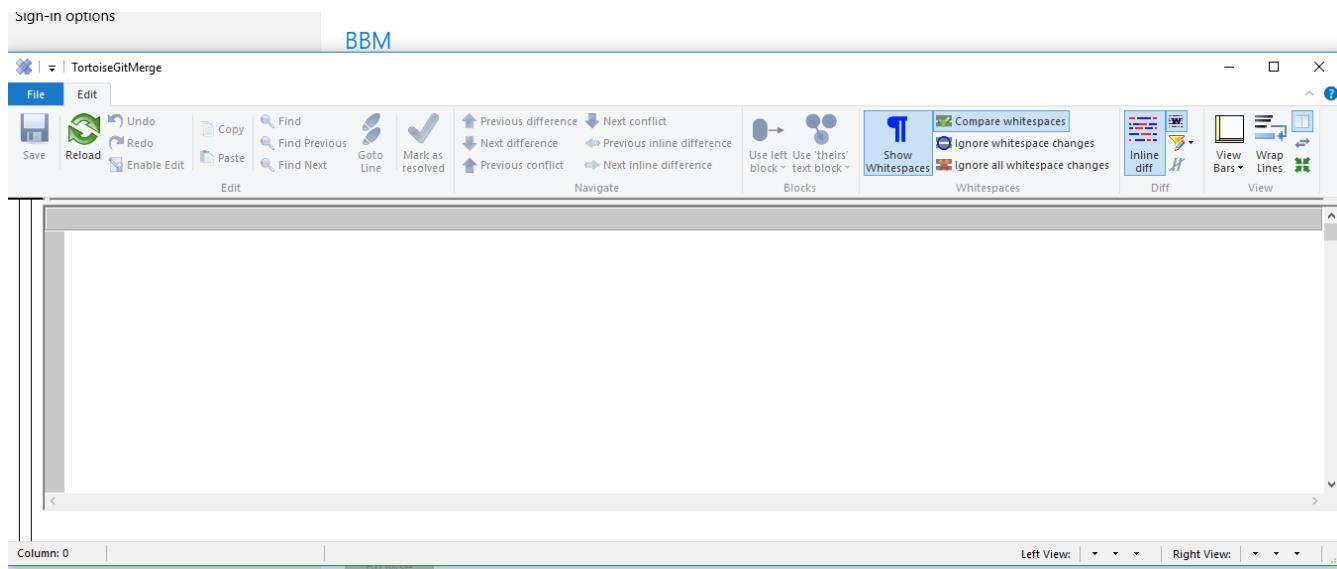
Configuramos un usuario y un email (igual que en Git Bash)



La siguiente hoja la dejamos por defecto y le damos a *Finish*:



Como se puede apreciar, Tortoisegit es un entorno gráfico



BIBLIOGRAFÍA

- <https://git-scm.com/book/es/v2/Inicio---Sobre-el-Control-de-Versiones-Acerca-del-Control-de-Versiones>
- <https://git-scm.com/docs/git-show>
- <https://git-scm.com/docs/git-reflog>
- <https://git-scm.com/docs/git-reset>
- <https://git-scm.com/docs/git-revert>
- <https://git-scm.com/docs/git-rebase>
- <https://git-scm.com/docs/git-stash>
- <http://dar10comyr.blogspot.com/2016/07/git-ramas-branch.html>
- <https://docs.microsoft.com/es-es/sql/ssms/download-sql-server-management-studio-ssms?view=sql-server-ver15#previous-versions>
- <https://www.microsoft.com/es-es/sql-server/sql-server-downloads>
- <https://medium.com/@jointdeveloper/sistemas-de-control-de-versiones-que-%C3%A9-son-y-por-que-%C3%A9-amarlos-24b6957e716e>
- <https://docs.microsoft.com/es-es/sql/database-engine/install-windows/rename-a-computer-that-hosts-a-stand-alone-instance-of-sql-server?view=sql-server-ver15>
- <https://docs.github.com/es/free-pro-team@latest/github/creating-cloning-and-archiving-repositories/about-readmes>
- <https://docs.github.com/es/free-pro-team@latest/github/using-git/ignoring-files>
- <https://www.pixelstech.net/article/1549115148-git-reset-vs-git-revert>
- <https://git-scm.com/book/es/v2/Ramificaciones-en-Git-%C2%BFQu%C3%A9-es-una-rama%3F>
- <https://docs.microsoft.com/es-es/sql/linux/quickstart-install-connect-ubuntu?view=sql-server-ver15>
- <https://oraxedatabase.blogspot.com/2019/09/como-desbloquear-una-cuenta-de-usuario.html>

<https://docs.microsoft.com/es-es/sql/relational-databases/replication/non-sql/manage-oracle-tablespaces?view=sql-server-ver15#:~:text=Un%20espacio%20de%20tabla%20es,archivos%20de%20Microsoft%20SQL%20Server%20.>

<https://www.sqlshack.com/sql-server-2019-on-linux-with-ubuntu-and-azure-data-studio/>

<https://database.guide/how-to-install-sqlcmd-bcp-on-ubuntu/>

<https://www.sqlshack.com/es/trabajar-con-la-linea-de-comando-de-sql-server-sqlcmd/>

<https://docs.microsoft.com/es-es/sql/connect/odbc/linux-mac/connecting-with-sqlcmd?view=sql-server-ver15>

<https://docs.microsoft.com/es-es/windows/deployment/windows-deployment-scenarios-and-tools>

<https://thesolving.com/es/sala-de-servidores/cuando-y-como-utilizar-sysprep/>

[https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2008-R2-and-2008/cc732351\(v=ws.10\)?redirectedfrom=MSDN](https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2008-R2-and-2008/cc732351(v=ws.10)?redirectedfrom=MSDN)

<https://docs.microsoft.com/es-es/sql/relational-databases/stored-procedures/stored-procedures-database-engine?view=sql-server-ver15>

<https://docs.microsoft.com/es-es/sql/relational-databases/databases/database-files-and-filegroups?view=sql-server-ver15>

<https://docs.microsoft.com/es-es/sql/relational-databases/blob/filestream-sql-server?view=sql-server-ver15>

<https://dba-presents.com/index.php/databases/sql-server/59-introduction-to-filestream>

<https://www.sqlshack.com/sql-server-filetable-the-next-generation-of-sql-filestream/>

<https://docs.microsoft.com/es-es/sql/relational-databases/blob/filetables-sql-server?view=sql-server-ver15>

<https://analisisyprogramacionoop.blogspot.com/2017/01/bases-de-datos-parcialmente->

[contenidas.html# :~:text=Bases%20de%20datos%20contenidas,describen%20una%20base%20de%20datos.](#)

<https://www.gpsos.es/2019/04/particionamiento-de-tablas-e-indices-t-sql-en-sql-server/>

<https://www.sqlshack.com/es/particionamiento-de-tablas-de-bases-de-datos-en-sql-server/>

<https://blogs.solidq.com/es/sql-server/particionado-en-sql-server-iii/>

<https://www.mssqltips.com/sqlservertip/4436/sql-server-2016-truncate-table-with-partitions/>

<https://www.sqlshack.com/es/tablas-temporales-sql-server/>

<https://docs.microsoft.com/en-us/sql/relational-databases/tables/querying-data-in-a-system-versioned-temporal-table?view=sql-server-2017>

<https://www.mssqltips.com/sqlservertip/3682/sql-server-2016-tsql-syntax-to-query-temporal-tables/>

[https://blog.comminus.hr/Comminus-Blog/October-2018-\(1\)/Temporal-Tables.aspx](https://blog.comminus.hr/Comminus-Blog/October-2018-(1)/Temporal-Tables.aspx)

<https://www.sqlshack.com/es/como-consultar-los-datos-en-tablas-temporales-de-version-del-sistema-en-sql-server/>

<https://www.mssqltips.com/sqlservertip/5436/options-to-retrieve-sql-server-temporal-table-and-history-data/>

<https://geeksencuentrena.com/windows/como-crear-una-instalacion-desatendida-de-windows-10/>

<https://www.cyclonis.com/es/what-is-efi-system-partition-can-you-delete-it/# :~:text=Una%20partici%C3%B3n%20de%20sistema%20EFI,sea%20Windows%20o%20Mac%20OS.>

<https://www.iperiusbackup.net/es/sysprep-clonacion-y-despliegue-de-instalaciones-de-windows/>

<https://thesolving.com/es/sala-de-servidores/cuando-y-como-utilizar-sysprep/>

<https://www.prajwaldesai.com/windows-10-adk-versions/>

<https://docs.microsoft.com/es-es/windows/deployment/windows-deployment-scenarios-and-tools>

<https://www.tenforums.com/tutorials/103428-create-winpe-usb-iso.html>

<https://todosobrewindows.com/descargar-autounattend-xml-para-windows-10/>

<https://github.com/MicrosoftDocs/sql-docs.es-es/blob/live/docs/relational-databases/in-memory-oltp/introduction-to-memory-optimized-tables.md>

<https://docs.microsoft.com/en-us/sql/relational-databases/in-memory-oltp/in-memory-oltp-in-memory-optimization?view=sql-server-2017>

<https://docs.microsoft.com/es-es/sql/t-sql/statements/alter-database-transact-sql-compatibility-level?view=sql-server-ver15>

<https://docs.microsoft.com/es-es/sql/relational-databases/in-memory-oltp/creating-a-memory-optimized-table-and-a-natively-compiled-stored-procedure?view=sql-server-ver15>

<https://www.sqlshack.com/how-to-create-and-customize-sql-server-templates/>

<https://mostafaelmasry.com/2015/03/25/durable-vs-non-durable-tables-in-memory-oltp/>

<https://docs.microsoft.com/es-es/sql/t-sql/statements/set-nocount-transact-sql?view=sql-server-ver15>

<https://www.mssqltips.com/sqlservertip/1070/simple-script-to-backup-all-sql-server-databases/>

<https://docs.microsoft.com/es-es/sql/relational-databases/stored-procedures/stored-procedures-database-engine?view=sql-server-ver15>

<https://www.adictosaltrabajo.com/2008/07/11/triggersql/>

<https://medievalstrucos.wordpress.com/2018/05/16/triggers-stored-procedure-transacciones-y-concurrencias-en-sql/>

<https://www.sqlservertutorial.net/sql-server-triggers/>

<https://docs.microsoft.com/es-es/sql/t-sql/statements/create-trigger-transact-sql?view=sql-server-ver15>

<https://docs.microsoft.com/es-es/sql/relational-databases/triggers/use-the-inserted-and-deleted-tables?view=sql-server-ver15>

<https://database.guide/create-a-last-modified-column-in-sql-server/>

