Herramientas de Programación 1

Tema Nº9:

Entity Framework (EF) - ORM

Indicador de logro Nº9:Desarrolla aplicaciones con ORM utilizando Entity Framework como tecnología de accesos a información en una base de datos.

**TEMA 01 Teoría de los**

Imagen que contiene Icono

Descripción generada automáticamente

**TEMA Nº9:**

SQL SERVER ENTITY FRAMEWORK

Base De Datos

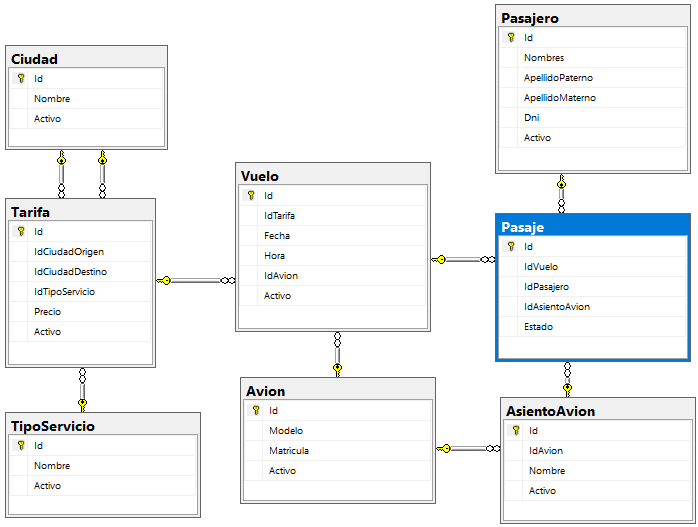
Ejemplo

Nuestra Base de Datos de ejemplo implementa la funcionalidad de venta de pasajes aéreos.

El nombre de la BD es: BD\_LATAM.

Trabajaremos con esta BD durante las semanas 9, 10 y 11.

Diagrama de la BD\_LATAM:



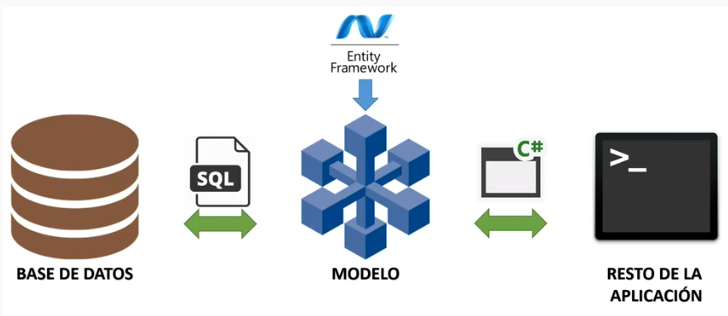
Antes de definir Entity Framework revisaremos sus orígenes, esto es, entender el concepto ORM (Mapeo de Objetos Relacionales).

ORM

Cuando se tiene una aplicación compleja se necesita de una BD, esta permitirá organizar los datos en tablas relacionadas entre sí con campos ya predefinidos en cada una de ellas. Los gestores que manejas este tipo de base de datos pueden ser SQL Server, Oracle entre otros.

El Modelo son un conjunto de Clases que nos trae la estructura relacional de cata tabla perteneciente a la BD, por ejemplo, una determinada Clase representa una tabla en específico donde los campos de esta pasan a ser las propiedades de la Clase.

El Modelo se encargara de generar todo el script SQL necesario para comunicarse con la BD simplificando las tareas básicas de acceso de datos para el programador.



El Modelo se encargará de generar todo el script SQL necesario para comunicarse con la BD simplificando las tareas básicas de acceso de datos para el programador.

El resto de la aplicación nunca tendrá acceso a la BD si no es a través del Modelo, en otras palabras, si la aplicación quiere acceder y operar sobre la BD (consultar, crear, modificar y eliminar datos) lo hará a través del Modelo. Y esta es la ventaja debido a que el Modelo estará escrito en el mismo lenguaje de programación que toda la aplicación (en nuestro caso C#).

Entity Framework

Entendido todo lo anterior Entity Framework es un ORM que nos permite generar el Modelo, ventajas:

* Facilidad y velocidad de uso.
* Absorción de las BD usadas.
* Seguridad de la capa a acceso a datos contra ataques.

Desventajas:

En entornos con gran carga de información, por agregar una capa más en el proceso, puede mermar en el rendimiento.

**Ejemplos:**

**RECURSOS**

1. Hardware

* Una computadora con 4 GB e RAM y procesador de 2.0 GHz.

1. Software

* Visual Studio Professional 2015 / 2019
* SQL Server 2016 / 2019

En los ejemplos se utilizó la versión Express 2016, puede usar las versiones comerciales o la Developer.

* SQL Server Management Studio (SSMS)

Es la herramienta cliente más usada para gestionar las BD de SQL Server.

**PROCEDIMIENTO**

**Laboratorio 9:**

9.1 SQL Server.

**Objetivos**

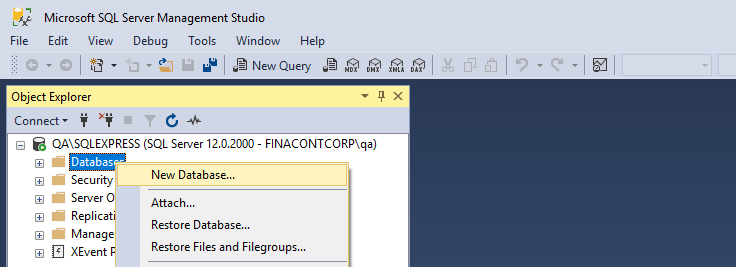
* Crear Base de Datos de ejemplo en SQL Server.
* Hacer consultas básicas DML.

**Introducción:**

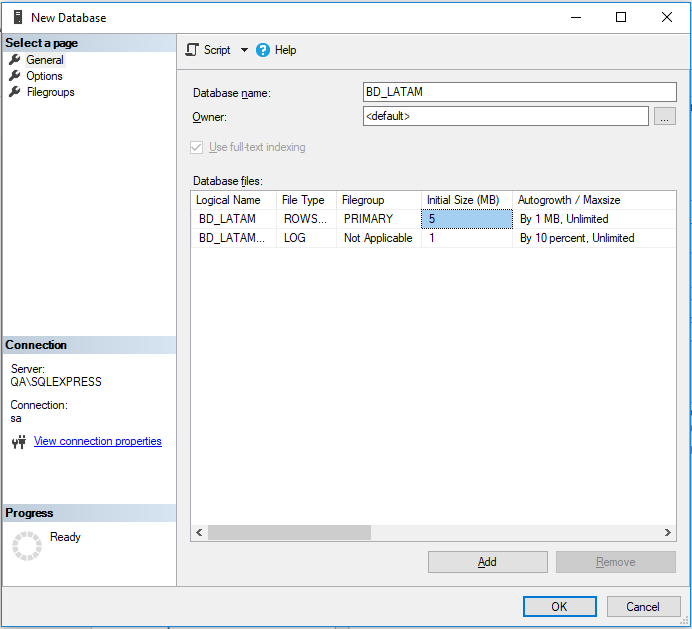
* Durante esta actividad, vamos a dar una rápida revisión a SQL Server, esto es:
  + Crear la BD de ejemplo: BD\_LATAM.
  + Comprender la estructura y relaciones entre las tablas de la BD\_LATAM.
  + Revisar los procedimientos almacenados de la BD\_LATAM.
* Desde Visual Studio aprenderemos a crear una conexión a la BD\_LATAM.

Paso 1. Crear la Base de Datos BD\_LATAM.

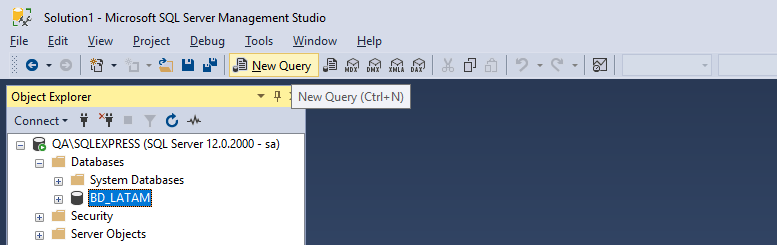
1. Abrimos SSMS.
2. Hacemos click derecho en la carpeta “Databases” y seleccionamos la opción “New Database…”



En la pantalla saliente ingresamos el nombre “BD\_LATAM” y presionamos en el botón “OK”.



1. Validamos que se ha creado la BD\_LATAM. Es una BD sin tablas, para crear las tablas vamos ejecutar un script sql, para ello primero presionamos la opción “New Query”



1. Copiamos el siguiente script (DDL):

USE [BD\_LATAM]

GO

/\*\*\*\*\*\* Object: Table [dbo].[AsientoAvion] Script Date: 24/08/2019 12:17:24 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[AsientoAvion](

[Id] [int] IDENTITY(1,1) NOT NULL,

[IdAvion] [int] NULL,

[Nombre] [nvarchar](10) NULL,

[Activo] [bit] NULL,

CONSTRAINT [PK\_AsientoAvion] PRIMARY KEY CLUSTERED

(

[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

/\*\*\*\*\*\* Object: Table [dbo].[Avion] Script Date: 24/08/2019 12:17:24 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Avion](

[Id] [int] IDENTITY(1,1) NOT NULL,

[Modelo] [nvarchar](50) NULL,

[Matricula] [nvarchar](50) NULL,

[Activo] [bit] NULL,

CONSTRAINT [PK\_Avion] PRIMARY KEY CLUSTERED

(

[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

/\*\*\*\*\*\* Object: Table [dbo].[Ciudad] Script Date: 24/08/2019 12:17:24 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Ciudad](

[Id] [int] IDENTITY(1,1) NOT NULL,

[Nombre] [nvarchar](50) NULL,

[Activo] [bit] NULL,

CONSTRAINT [PK\_Ciudad] PRIMARY KEY CLUSTERED

(

[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

/\*\*\*\*\*\* Object: Table [dbo].[Pasaje] Script Date: 24/08/2019 12:17:24 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Pasaje](

[Id] [int] IDENTITY(1,1) NOT NULL,

[IdVuelo] [int] NULL,

[IdPasajero] [int] NULL,

[IdAsientoAvion] [int] NULL,

[Estado] [nvarchar](10) NULL,

CONSTRAINT [PK\_Pasaje] PRIMARY KEY CLUSTERED

(

[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

/\*\*\*\*\*\* Object: Table [dbo].[Pasajero] Script Date: 24/08/2019 12:17:24 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Pasajero](

[Id] [int] IDENTITY(1,1) NOT NULL,

[Nombres] [nvarchar](50) NULL,

[ApellidoPaterno] [nvarchar](50) NULL,

[ApellidoMaterno] [nvarchar](50) NULL,

[Dni] [nvarchar](10) NULL,

[Activo] [bit] NULL,

CONSTRAINT [PK\_Pasajero] PRIMARY KEY CLUSTERED

(

[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

/\*\*\*\*\*\* Object: Table [dbo].[Tarifa] Script Date: 24/08/2019 12:17:25 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Tarifa](

[Id] [int] IDENTITY(1,1) NOT NULL,

[IdCiudadOrigen] [int] NULL,

[IdCiudadDestino] [int] NULL,

[IdTipoServicio] [int] NULL,

[Precio] [decimal](10, 2) NULL,

[Activo] [bit] NULL,

CONSTRAINT [PK\_Tarifa] PRIMARY KEY CLUSTERED

(

[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

/\*\*\*\*\*\* Object: Table [dbo].[TipoServicio] Script Date: 24/08/2019 12:17:25 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[TipoServicio](

[Id] [int] IDENTITY(1,1) NOT NULL,

[Nombre] [nvarchar](50) NULL,

[Activo] [bit] NULL,

CONSTRAINT [PK\_TipoServicio] PRIMARY KEY CLUSTERED

(

[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

/\*\*\*\*\*\* Object: Table [dbo].[Vuelo] Script Date: 24/08/2019 12:17:25 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Vuelo](

[Id] [int] IDENTITY(1,1) NOT NULL,

[IdTarifa] [int] NULL,

[Fecha] [nvarchar](10) NULL,

[Hora] [nvarchar](10) NULL,

[IdAvion] [int] NULL,

[Activo] [bit] NULL,

CONSTRAINT [PK\_Vuelo] PRIMARY KEY CLUSTERED

(

[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

ALTER TABLE [dbo].[AsientoAvion] WITH CHECK ADD CONSTRAINT [FK\_AsientoAvion\_Avion] FOREIGN KEY([IdAvion])

REFERENCES [dbo].[Avion] ([Id])

GO

ALTER TABLE [dbo].[AsientoAvion] CHECK CONSTRAINT [FK\_AsientoAvion\_Avion]

GO

ALTER TABLE [dbo].[Pasaje] WITH CHECK ADD CONSTRAINT [FK\_Pasaje\_AsientoAvion] FOREIGN KEY([IdAsientoAvion])

REFERENCES [dbo].[AsientoAvion] ([Id])

GO

ALTER TABLE [dbo].[Pasaje] CHECK CONSTRAINT [FK\_Pasaje\_AsientoAvion]

GO

ALTER TABLE [dbo].[Pasaje] WITH CHECK ADD CONSTRAINT [FK\_Pasaje\_Pasajero1] FOREIGN KEY([IdPasajero])

REFERENCES [dbo].[Pasajero] ([Id])

GO

ALTER TABLE [dbo].[Pasaje] CHECK CONSTRAINT [FK\_Pasaje\_Pasajero1]

GO

ALTER TABLE [dbo].[Pasaje] WITH CHECK ADD CONSTRAINT [FK\_Pasaje\_Vuelo] FOREIGN KEY([IdVuelo])

REFERENCES [dbo].[Vuelo] ([Id])

GO

ALTER TABLE [dbo].[Pasaje] CHECK CONSTRAINT [FK\_Pasaje\_Vuelo]

GO

ALTER TABLE [dbo].[Tarifa] WITH CHECK ADD CONSTRAINT [FK\_Tarifa\_Ciudad] FOREIGN KEY([IdCiudadOrigen])

REFERENCES [dbo].[Ciudad] ([Id])

GO

ALTER TABLE [dbo].[Tarifa] CHECK CONSTRAINT [FK\_Tarifa\_Ciudad]

GO

ALTER TABLE [dbo].[Tarifa] WITH CHECK ADD CONSTRAINT [FK\_Tarifa\_Ciudad1] FOREIGN KEY([IdCiudadDestino])

REFERENCES [dbo].[Ciudad] ([Id])

GO

ALTER TABLE [dbo].[Tarifa] CHECK CONSTRAINT [FK\_Tarifa\_Ciudad1]

GO

ALTER TABLE [dbo].[Tarifa] WITH CHECK ADD CONSTRAINT [FK\_Tarifa\_TipoServicio] FOREIGN KEY([IdTipoServicio])

REFERENCES [dbo].[TipoServicio] ([Id])

GO

ALTER TABLE [dbo].[Tarifa] CHECK CONSTRAINT [FK\_Tarifa\_TipoServicio]

GO

ALTER TABLE [dbo].[Vuelo] WITH CHECK ADD CONSTRAINT [FK\_Vuelo\_Avion] FOREIGN KEY([IdAvion])

REFERENCES [dbo].[Avion] ([Id])

GO

ALTER TABLE [dbo].[Vuelo] CHECK CONSTRAINT [FK\_Vuelo\_Avion]

GO

ALTER TABLE [dbo].[Vuelo] WITH CHECK ADD CONSTRAINT [FK\_Vuelo\_Tarifa] FOREIGN KEY([IdTarifa])

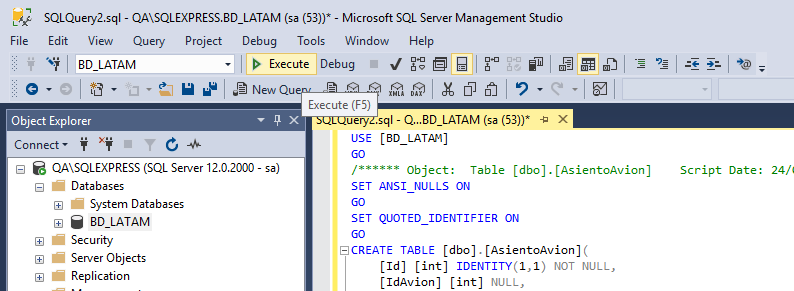
REFERENCES [dbo].[Tarifa] ([Id])

GO

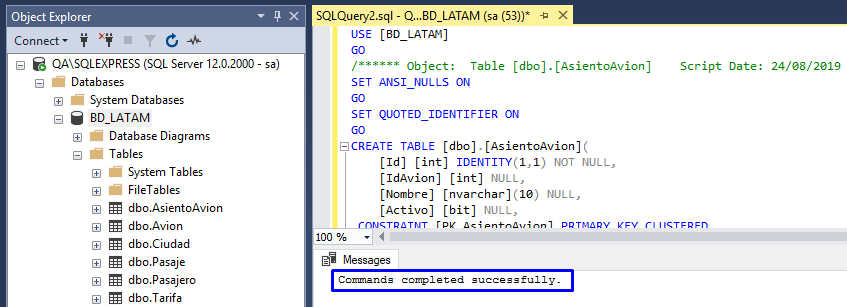
ALTER TABLE [dbo].[Vuelo] CHECK CONSTRAINT [FK\_Vuelo\_Tarifa]

GO

1. Ahora presionamos en la opción “Execute”



1. Validamos que el script se ha ejecutado correctamente:



1. Hasta este instante hemos creado las tablas y sus relaciones, sin embargo no almacena ninguna información, la ejecución del siguiente script (DML) nos facilitara la carga inicial de datos:

USE [BD\_LATAM]

GO

SET IDENTITY\_INSERT [dbo].[Avion] ON

GO

INSERT [dbo].[Avion] ([Id], [Modelo], [Matricula], [Activo]) VALUES (1, N'AIRBUS A300', NULL, 0)

GO

INSERT [dbo].[Avion] ([Id], [Modelo], [Matricula], [Activo]) VALUES (2, N'BOEING 737', NULL, 1)

GO

INSERT [dbo].[Avion] ([Id], [Modelo], [Matricula], [Activo]) VALUES (3, N'AIRBUS A340', NULL, 1)

GO

INSERT [dbo].[Avion] ([Id], [Modelo], [Matricula], [Activo]) VALUES (4, N'AIRBUS A350', NULL, 1)

GO

INSERT [dbo].[Avion] ([Id], [Modelo], [Matricula], [Activo]) VALUES (5, N'BOEING 747', NULL, 1)

GO

SET IDENTITY\_INSERT [dbo].[Avion] OFF

GO

SET IDENTITY\_INSERT [dbo].[AsientoAvion] ON

GO

INSERT [dbo].[AsientoAvion] ([Id], [IdAvion], [Nombre], [Activo]) VALUES (1, 1, N'A1', 1)

GO

INSERT [dbo].[AsientoAvion] ([Id], [IdAvion], [Nombre], [Activo]) VALUES (2, 1, N'A2', 1)

GO

INSERT [dbo].[AsientoAvion] ([Id], [IdAvion], [Nombre], [Activo]) VALUES (3, 1, N'A3', 1)

GO

INSERT [dbo].[AsientoAvion] ([Id], [IdAvion], [Nombre], [Activo]) VALUES (4, 2, N'B1', 1)

GO

INSERT [dbo].[AsientoAvion] ([Id], [IdAvion], [Nombre], [Activo]) VALUES (5, 2, N'B2', 1)

GO

INSERT [dbo].[AsientoAvion] ([Id], [IdAvion], [Nombre], [Activo]) VALUES (6, 2, N'B3', 1)

GO

INSERT [dbo].[AsientoAvion] ([Id], [IdAvion], [Nombre], [Activo]) VALUES (7, 3, N'C1', 1)

GO

INSERT [dbo].[AsientoAvion] ([Id], [IdAvion], [Nombre], [Activo]) VALUES (8, 3, N'C2', 1)

GO

INSERT [dbo].[AsientoAvion] ([Id], [IdAvion], [Nombre], [Activo]) VALUES (9, 3, N'C3', 1)

GO

INSERT [dbo].[AsientoAvion] ([Id], [IdAvion], [Nombre], [Activo]) VALUES (10, 4, N'D1', 1)

GO

INSERT [dbo].[AsientoAvion] ([Id], [IdAvion], [Nombre], [Activo]) VALUES (11, 4, N'D2', 1)

GO

INSERT [dbo].[AsientoAvion] ([Id], [IdAvion], [Nombre], [Activo]) VALUES (12, 4, N'D3', 1)

GO

INSERT [dbo].[AsientoAvion] ([Id], [IdAvion], [Nombre], [Activo]) VALUES (13, 5, N'E1', 1)

GO

INSERT [dbo].[AsientoAvion] ([Id], [IdAvion], [Nombre], [Activo]) VALUES (14, 5, N'E2', 1)

GO

INSERT [dbo].[AsientoAvion] ([Id], [IdAvion], [Nombre], [Activo]) VALUES (15, 5, N'E3', 1)

GO

SET IDENTITY\_INSERT [dbo].[AsientoAvion] OFF

GO

SET IDENTITY\_INSERT [dbo].[Pasajero] ON

GO

INSERT [dbo].[Pasajero] ([Id], [Nombres], [ApellidoPaterno], [ApellidoMaterno], [Dni], [Activo]) VALUES (1, N'JUAN', N'PEREZ', N'PINO', N'01020304', 1)

GO

INSERT [dbo].[Pasajero] ([Id], [Nombres], [ApellidoPaterno], [ApellidoMaterno], [Dni], [Activo]) VALUES (2, N'ISABEL', N'RODRIGUEZ', N'LOZANO', N'16164789', 1)

GO

INSERT [dbo].[Pasajero] ([Id], [Nombres], [ApellidoPaterno], [ApellidoMaterno], [Dni], [Activo]) VALUES (3, N'LUCY', N'MATIAS', N'POVIS', N'88750102', 1)

GO

INSERT [dbo].[Pasajero] ([Id], [Nombres], [ApellidoPaterno], [ApellidoMaterno], [Dni], [Activo]) VALUES (4, N'PEDRO', N'VARGAS', N'JARAMILLO', N'56551487', 1)

GO

SET IDENTITY\_INSERT [dbo].[Pasajero] OFF

GO

SET IDENTITY\_INSERT [dbo].[Ciudad] ON

GO

INSERT [dbo].[Ciudad] ([Id], [Nombre], [Activo]) VALUES (1, N'LIMA', 1)

GO

INSERT [dbo].[Ciudad] ([Id], [Nombre], [Activo]) VALUES (2, N'CUZCO', 1)

GO

INSERT [dbo].[Ciudad] ([Id], [Nombre], [Activo]) VALUES (3, N'TRUJILLO', 1)

GO

INSERT [dbo].[Ciudad] ([Id], [Nombre], [Activo]) VALUES (4, N'IQUITOS', 1)

GO

INSERT [dbo].[Ciudad] ([Id], [Nombre], [Activo]) VALUES (5, N'AREQUIPA', 1)

GO

INSERT [dbo].[Ciudad] ([Id], [Nombre], [Activo]) VALUES (6, N'TACNA', 0)

GO

SET IDENTITY\_INSERT [dbo].[Ciudad] OFF

GO

SET IDENTITY\_INSERT [dbo].[TipoServicio] ON

GO

INSERT [dbo].[TipoServicio] ([Id], [Nombre], [Activo]) VALUES (1, N'VIP', 1)

GO

INSERT [dbo].[TipoServicio] ([Id], [Nombre], [Activo]) VALUES (2, N'ECONOMICO', 1)

GO

INSERT [dbo].[TipoServicio] ([Id], [Nombre], [Activo]) VALUES (3, N'TURISTA', 1)

GO

INSERT [dbo].[TipoServicio] ([Id], [Nombre], [Activo]) VALUES (4, N'INVITADO', 0)

GO

SET IDENTITY\_INSERT [dbo].[TipoServicio] OFF

GO

SET IDENTITY\_INSERT [dbo].[Tarifa] ON

GO

INSERT [dbo].[Tarifa] ([Id], [IdCiudadOrigen], [IdCiudadDestino], [IdTipoServicio], [Precio], [Activo]) VALUES (1, 1, 2, 1, CAST(300.00 AS Decimal(10, 2)), 1)

GO

INSERT [dbo].[Tarifa] ([Id], [IdCiudadOrigen], [IdCiudadDestino], [IdTipoServicio], [Precio], [Activo]) VALUES (2, 1, 2, 2, CAST(200.00 AS Decimal(10, 2)), 1)

GO

INSERT [dbo].[Tarifa] ([Id], [IdCiudadOrigen], [IdCiudadDestino], [IdTipoServicio], [Precio], [Activo]) VALUES (3, 1, 2, 3, CAST(250.00 AS Decimal(10, 2)), 1)

GO

INSERT [dbo].[Tarifa] ([Id], [IdCiudadOrigen], [IdCiudadDestino], [IdTipoServicio], [Precio], [Activo]) VALUES (4, 2, 1, 1, CAST(305.00 AS Decimal(10, 2)), 1)

GO

INSERT [dbo].[Tarifa] ([Id], [IdCiudadOrigen], [IdCiudadDestino], [IdTipoServicio], [Precio], [Activo]) VALUES (5, 2, 1, 2, CAST(205.00 AS Decimal(10, 2)), 1)

GO

INSERT [dbo].[Tarifa] ([Id], [IdCiudadOrigen], [IdCiudadDestino], [IdTipoServicio], [Precio], [Activo]) VALUES (6, 2, 1, 3, CAST(255.00 AS Decimal(10, 2)), 1)

GO

SET IDENTITY\_INSERT [dbo].[Tarifa] OFF

GO

SET IDENTITY\_INSERT [dbo].[Vuelo] ON

GO

INSERT [dbo].[Vuelo] ([Id], [IdTarifa], [Fecha], [Hora], [IdAvion], [Activo]) VALUES (1, 1, N'30-10-2019', N'18:30', 1, 1)

GO

INSERT [dbo].[Vuelo] ([Id], [IdTarifa], [Fecha], [Hora], [IdAvion], [Activo]) VALUES (2, 5, N'01-11-2019', N'05:00', 5, 1)

GO

INSERT [dbo].[Vuelo] ([Id], [IdTarifa], [Fecha], [Hora], [IdAvion], [Activo]) VALUES (3, 2, N'28-10-2019', N'00:00', 2, 1)

GO

INSERT [dbo].[Vuelo] ([Id], [IdTarifa], [Fecha], [Hora], [IdAvion], [Activo]) VALUES (4, 1, N'27-11-2019', N'12:00', 3, 1)

GO

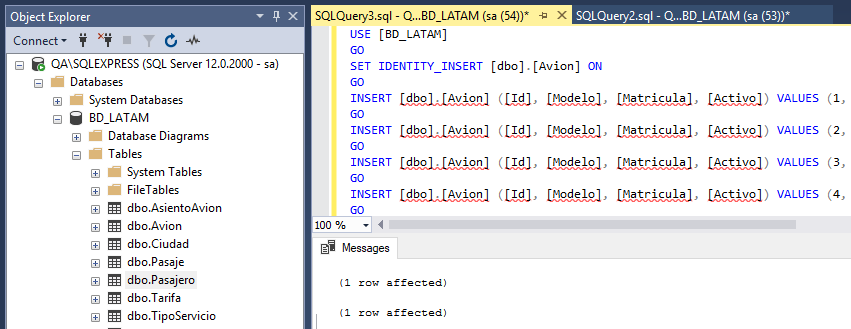
INSERT [dbo].[Vuelo] ([Id], [IdTarifa], [Fecha], [Hora], [IdAvion], [Activo]) VALUES (5, 3, N'26-10-2019', N'23:45', 4, 1)

GO

SET IDENTITY\_INSERT [dbo].[Vuelo] OFF

GO

1. Validamos que el script se ha ejecutado correctamente (no debe existir ningún mensaje de error):



1. Realizamos algunas consultas para validar la existencia de datos, por ejemplo, para ver que Tarifas tenemos registrado en la BD ejecutamos:

SELECT T.Id, T.IdCiudadOrigen, C1.Nombre AS 'Origen',

T.IdCiudadDestino, C2.Nombre AS 'Destino',

T.IdTipoServicio, TS.Nombre, T.Precio, IIF(T.Activo > 0, 'SI', 'NO') AS 'Activo'

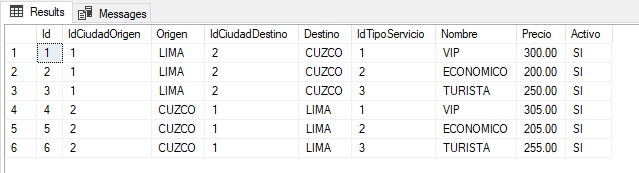
FROM Tarifa T

INNER JOIN Ciudad C1 ON T.IdCiudadOrigen = C1.Id

INNER JOIN Ciudad C2 ON T.IdCiudadDestino = C2.Id

INNER JOIN TipoServicio TS ON T.IdTipoServicio = TS.id;

1. Debemos obtener:



11.2 Entity Framework en Visual Studio.

Objetivos

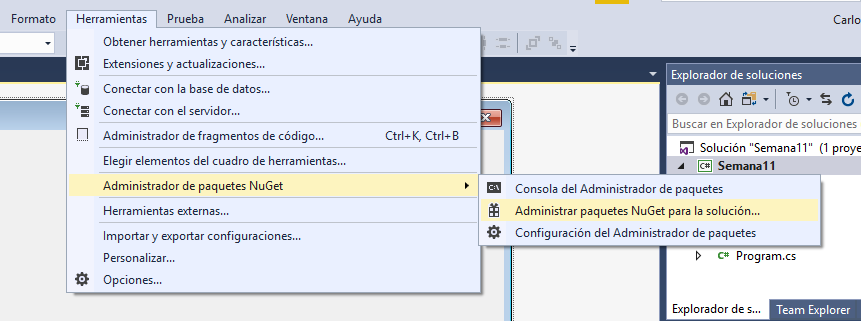
* Instalar Entity Framework en Visual Studio.

Introducción:

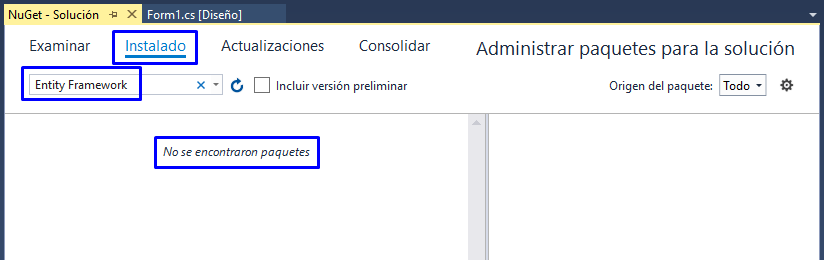
* Durante esta actividad, vamos a dar validar si ya tenemos instalado Entity Framework, si no lo tenemos instalado pasamos a instalarlo.

Paso 1. Validar si tenemos instalado Entity Framework.

1. Abrimos Visual Studio.
2. Creamos el proyecto Semana11.
3. Seleccionamos la opción “Administrar paquetes Nuget para la solución…”

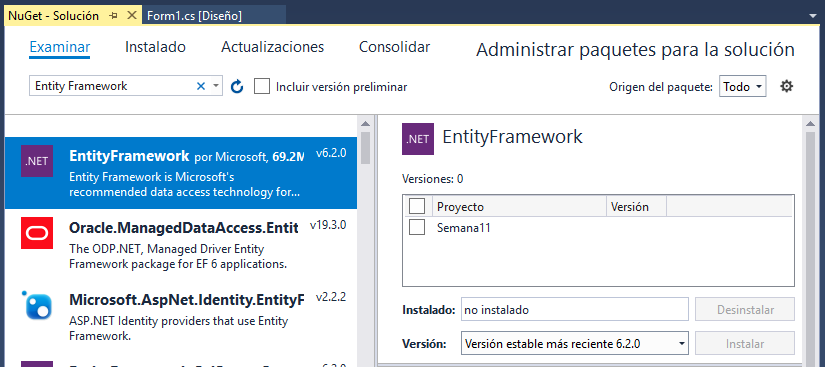


1. En la pestaña Instalado, dentro del cuadro de texto de búsqueda ponemos “Entity Framework” y presionamos la tecla Enter, si se muestra el mensaje que indica que no está instalado pasaremos a instalarlo.

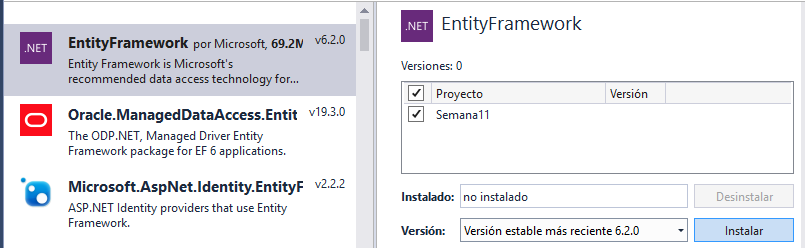


Paso 2. Instalar Entity Framework.

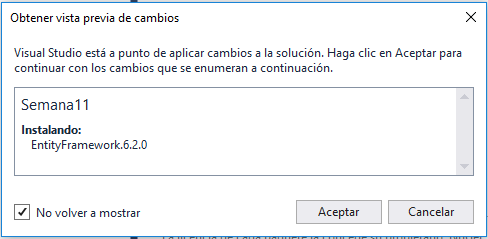
1. Seleccionamos la pestaña “Examina”, dentro del cuadro de texto de búsqueda ponemos “Entity Framework” y presionamos la tecla Enter.
2. De la lista de paquetes encontrados seleccionamos el primero “EntityFramework”:



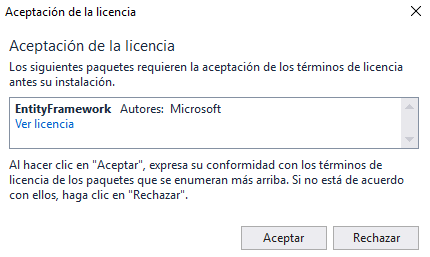
1. Hacemos check en la casilla “Semana11”, verificamos que la versión sea la reciente más estable y presionamos el botón “Instalar”.



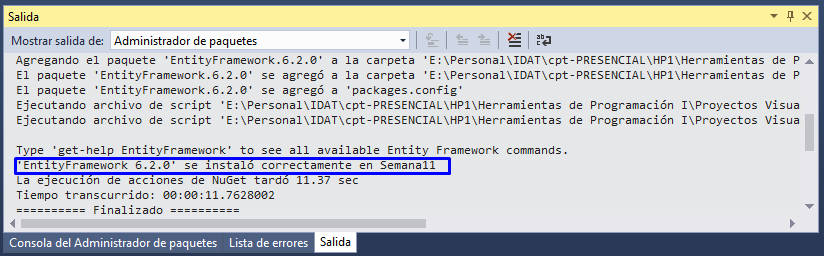
1. En el formulario que se muestra presionamos el botón “Aceptar”.



1. En el formulario que se muestra presionamos el botón “Aceptar”.



1. Validamos que el paquete de Entity Framework se ha instalado correctamente:



11.3 Orígenes de datos en Visual Studio.

Objetivos

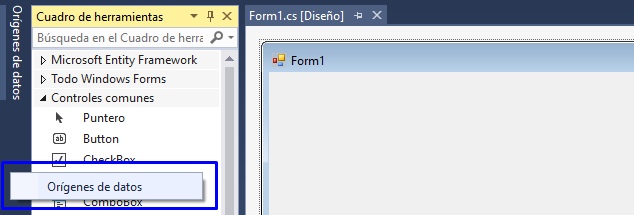
* Crear en Visual Studio un origen de datos a una BD SQL Server.

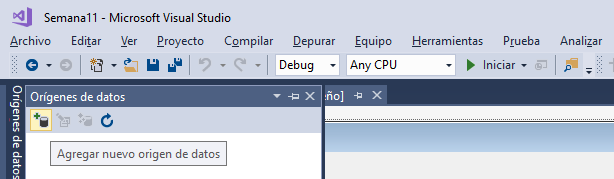
Introducción:

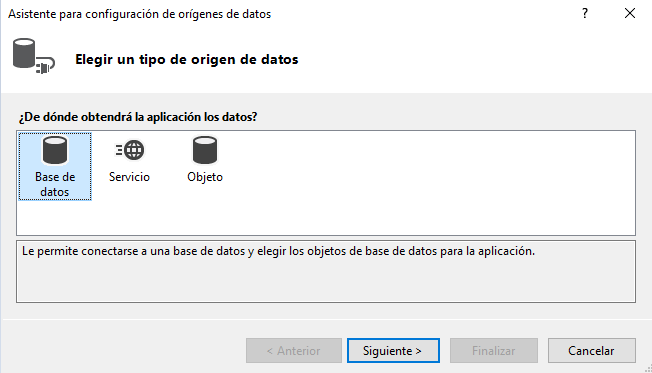
* Durante esta actividad, vamos aprender a crear una conexión a una BD SQL Server.

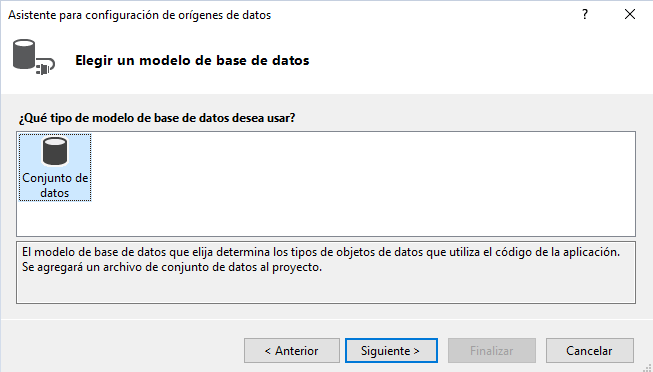
Paso 1. Crear una conexión a la BD BD\_LATAM.

1. En un lugar vacío de la franja izquierda hacemos click derecho y seleccionamos la opción “Orígenes de Datos”

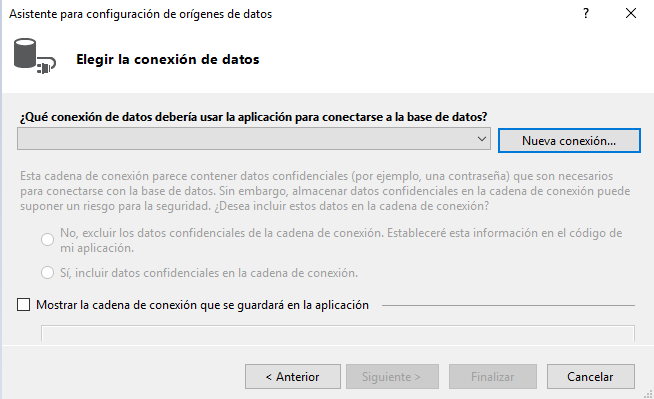




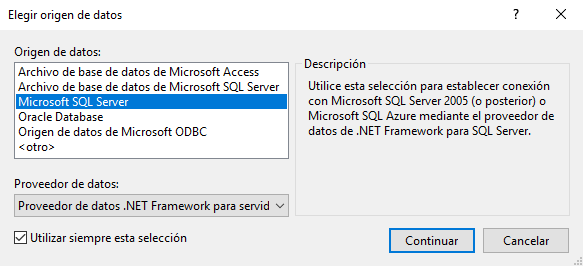
1. Seleccionamos la opción “Agregar nuevo origen de datos”
2. Seleccionamos “Base de datos” y presionamos en el botón “Siguiente >”



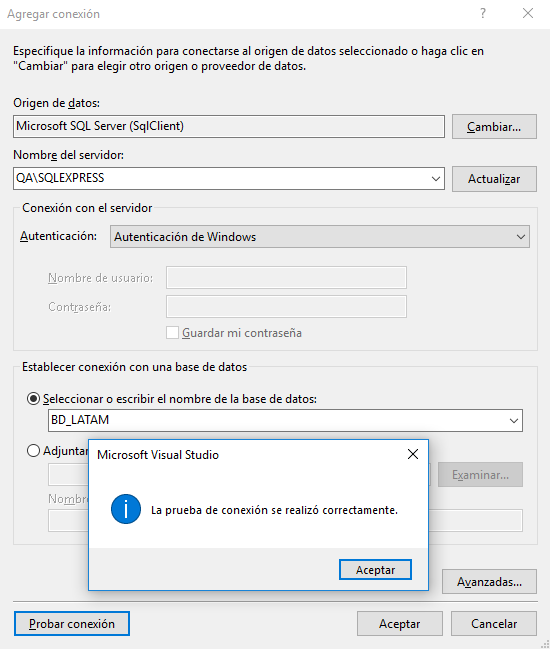
1. Seleccionamos “Conjunto de datos” y presionamos en el botón “Siguiente >”



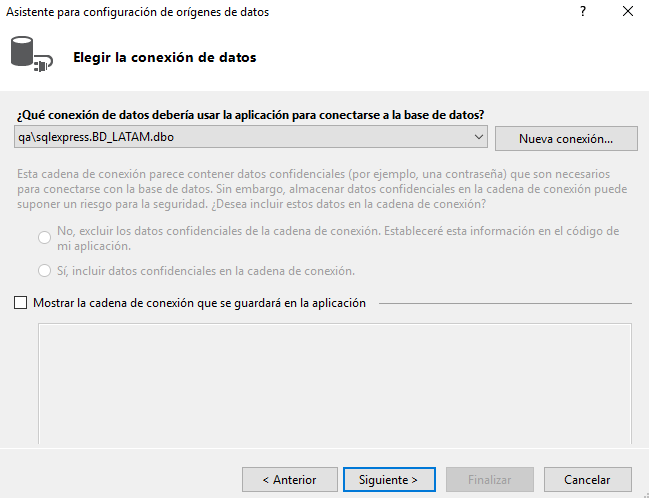
1. Seleccionamos el botón “Nueva conexión…”



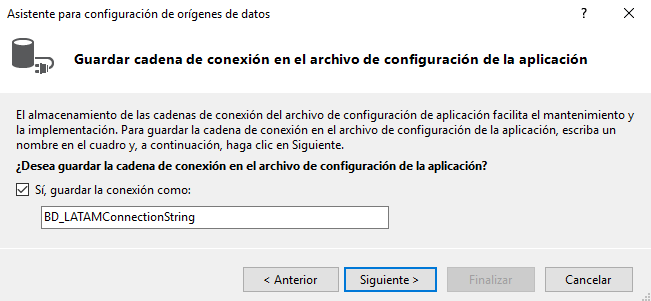
1. Seleccionamos las opciones que se muestran y presionamos el botón “Continuar”



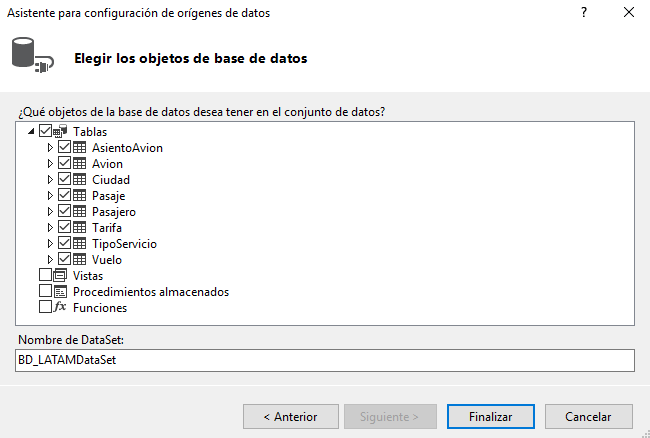
1. Seleccionamos las opciones que se muestran, probamos la conexión y presionamos el botón “Aceptar”



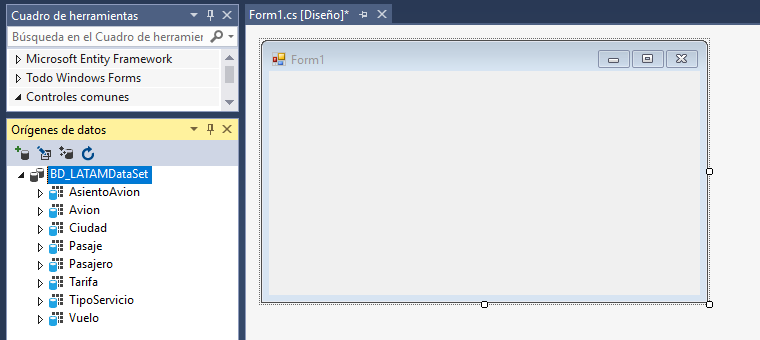
1. Presionamos en el botón “Siguiente >”



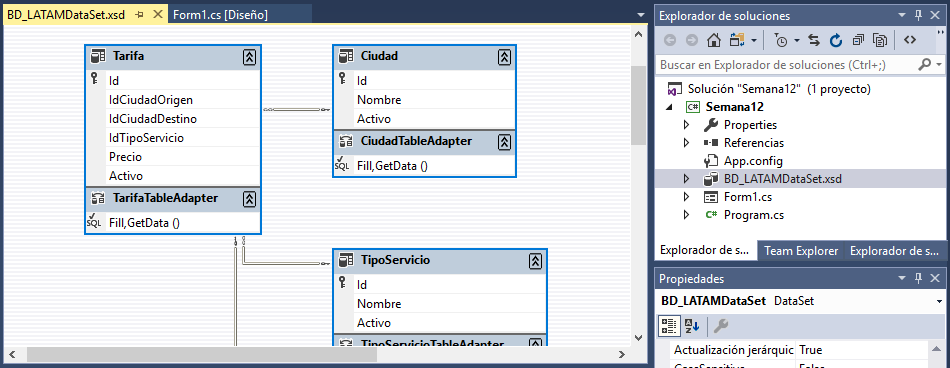
1. Presionamos en el botón “Siguiente >”



1. Como por el momento solo tenemos tablas en nuestra BD, seleccionamos todas las tablas de la BD\_LATAM y presionamos el botón “Finalizar”.



1. En el panel de “Orígenes de datos” ya podemos visualizar la nueva conexión.



1. Si hacemos doble click en BD\_LATAMDataSet.xsd visualizaremos las tablas que hemos seleccionado para el origen de datos.

Esta conexión la usaremos para conectar Entity Framework a la BD\_LATAM.

**Actividad:**

Revisa y analiza los siguientes enlaces que se encuentran en la plataforma virtual, luego desarrolla las actividades propuestas:

**ORM.**

<https://www.youtube.com/watch?v=kBZjaabLUtk>

**Entity Framework.**

<https://docs.microsoft.com/es-es/dotnet/framework/data/adonet/ef/overview>

1. Explicar y dar ejemplos de cinco sentencias SQL de tipo DDL (Lenguaje de definición de datos).
2. Explicar y dar ejemplos de cinco sentencias SQL de tipo DML (Lenguaje de manipulación de datos).
3. En la base de datos BD\_LATAM crear un procedimiento almacenado (SP) que devuelva el detalle de las tarifas activas. El SP debe recibir como parámetros:

* Ciudad de Origen
* Ciudad de Destino
* Tipo de servicio

1. En la base de datos BD\_LATAM crear un procedimiento almacenado (SP) que devuelva el detalle de los vuelos activos. El SP debe recibir como parámetros:

* Ciudad de Origen
* Ciudad de Destino
* Fecha