Herramientas de Programación 2

Tema Nº1:MVC

Indicador de logro Nº1:Implementa el acceso transparente a bases de datos en sus aplicaciones utilizando los softwares aprendidos en clase.

**TEMA 01 Teoría de los**

Imagen que contiene Icono

Descripción generada automáticamente

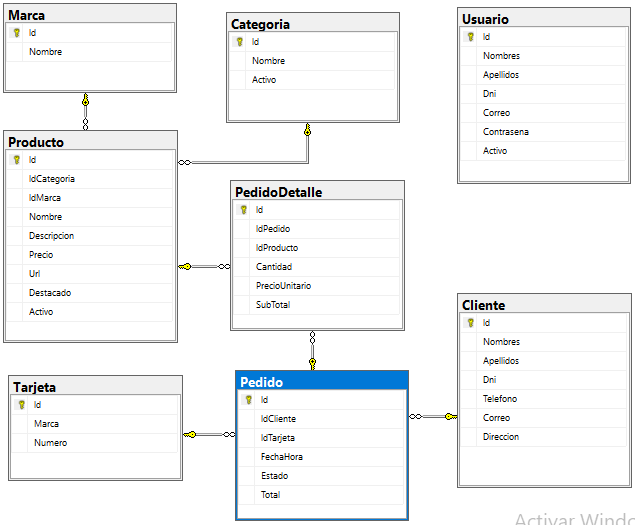
**TEMA Nº1:**

MVC

**Subtema 1.1:**

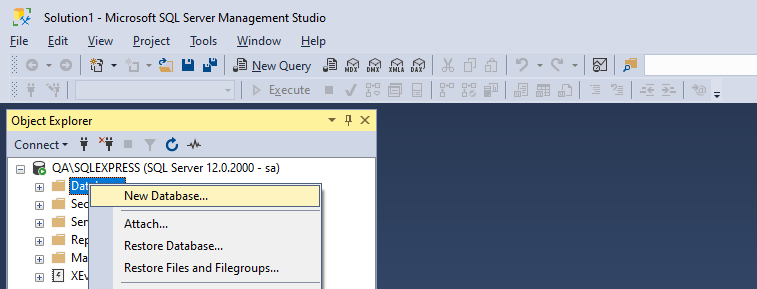
SQL SERVER

Estructura de la Base de Datos (BD) que servirá como referencia en los siguientes talleres.

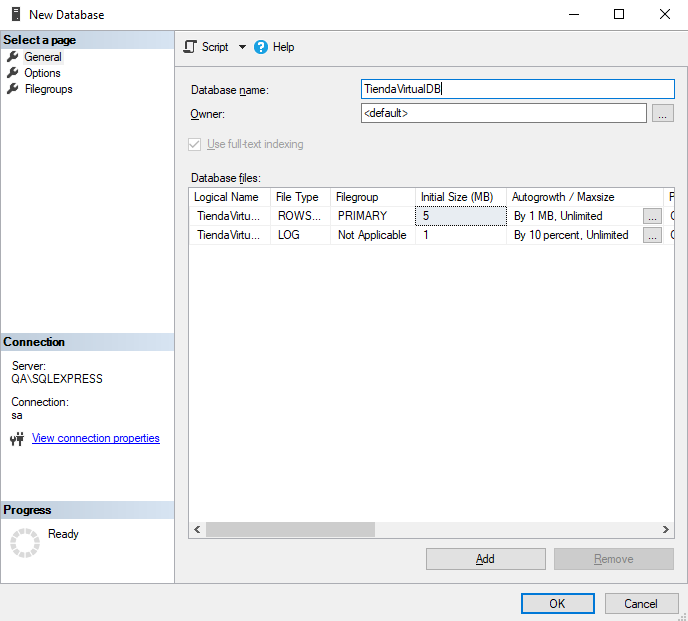


Paso 1. Crear la Base de Datos ***TiendaVirtualDB*** en SQL Server.

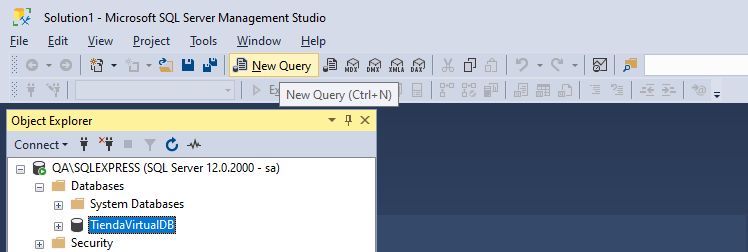
1. Abrimos SSMS.
2. Hacemos click derecho en la carpeta ***Databases*** y seleccionamos la opción: ***New Database…***



1. En la pantalla saliente ingresamos el nombre ***TiendaVirtualDB*** y presionamos en el botón ***OK***.



1. Validamos que se ha creado la BD ***TiendaVirtualDB***. 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 [TiendaVirtualDB]

GO

/\*\*\*\*\*\* Object: Table [dbo].[Categoria] \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Categoria](

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

[Nombre] [nvarchar](50) NULL,

[Activo] [bit] NULL,

CONSTRAINT [PK\_Categoria] 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].[Cliente] \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Cliente](

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

[Nombres] [nvarchar](100) NULL,

[Apellidos] [nvarchar](100) NULL,

[Dni] [nvarchar](8) NULL,

[Telefono] [nvarchar](50) NULL,

[Correo] [nvarchar](50) NULL,

[Direccion] [nvarchar](max) NULL,

CONSTRAINT [PK\_Cliente] 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] TEXTIMAGE\_ON [PRIMARY]

GO

/\*\*\*\*\*\* Object: Table [dbo].[Marca] \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Marca](

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

[Nombre] [nvarchar](50) NULL,

CONSTRAINT [PK\_Marca] 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].[Pedido] \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Pedido](

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

[IdCliente] [int] NULL,

[IdTarjeta] [int] NULL,

[FechaHora] [datetime] NULL,

[Estado] [nvarchar](50) NULL,

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

CONSTRAINT [PK\_Pedido] 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].[PedidoDetalle] \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[PedidoDetalle](

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

[IdPedido] [int] NULL,

[IdProducto] [int] NULL,

[Cantidad] [int] NULL,

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

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

CONSTRAINT [PK\_PedidoDetalle] 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].[Producto] \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Producto](

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

[IdCategoria] [int] NULL,

[IdMarca] [int] NULL,

[Nombre] [nvarchar](100) NULL,

[Descripcion] [nvarchar](max) NULL,

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

[Url] [nvarchar](max) NULL,

[Destacado] [bit] NULL,

[Activo] [bit] NULL,

CONSTRAINT [PK\_Producto] 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] TEXTIMAGE\_ON [PRIMARY]

GO

/\*\*\*\*\*\* Object: Table [dbo].[Tarjeta] \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Tarjeta](

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

[Marca] [nvarchar](50) NULL,

[Numero] [nvarchar](50) NULL,

CONSTRAINT [PK\_Tarjeta] 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].[Usuario] \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Usuario](

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

[Nombres] [nvarchar](50) NULL,

[Apellidos] [nvarchar](50) NULL,

[Dni] [nvarchar](8) NULL,

[Correo] [nvarchar](50) NULL,

[Contrasena] [nvarchar](100) NULL,

[Activo] [bit] NULL,

CONSTRAINT [PK\_Usuario] 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].[Pedido] WITH CHECK ADD CONSTRAINT [FK\_Pedido\_Cliente] FOREIGN KEY([IdCliente])

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

GO

ALTER TABLE [dbo].[Pedido] CHECK CONSTRAINT [FK\_Pedido\_Cliente]

GO

ALTER TABLE [dbo].[Pedido] WITH CHECK ADD CONSTRAINT [FK\_Pedido\_Tarjeta] FOREIGN KEY([IdTarjeta])

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

GO

ALTER TABLE [dbo].[Pedido] CHECK CONSTRAINT [FK\_Pedido\_Tarjeta]

GO

ALTER TABLE [dbo].[PedidoDetalle] WITH CHECK ADD CONSTRAINT [FK\_PedidoDetalle\_Pedido] FOREIGN KEY([IdPedido])

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

GO

ALTER TABLE [dbo].[PedidoDetalle] CHECK CONSTRAINT [FK\_PedidoDetalle\_Pedido]

GO

ALTER TABLE [dbo].[PedidoDetalle] WITH CHECK ADD CONSTRAINT [FK\_PedidoDetalle\_Producto] FOREIGN KEY([IdProducto])

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

GO

ALTER TABLE [dbo].[PedidoDetalle] CHECK CONSTRAINT [FK\_PedidoDetalle\_Producto]

GO

ALTER TABLE [dbo].[Producto] WITH CHECK ADD CONSTRAINT [FK\_Producto\_Categoria] FOREIGN KEY([IdCategoria])

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

GO

ALTER TABLE [dbo].[Producto] CHECK CONSTRAINT [FK\_Producto\_Categoria]

GO

ALTER TABLE [dbo].[Producto] WITH CHECK ADD CONSTRAINT [FK\_Producto\_Marca] FOREIGN KEY([IdMarca])

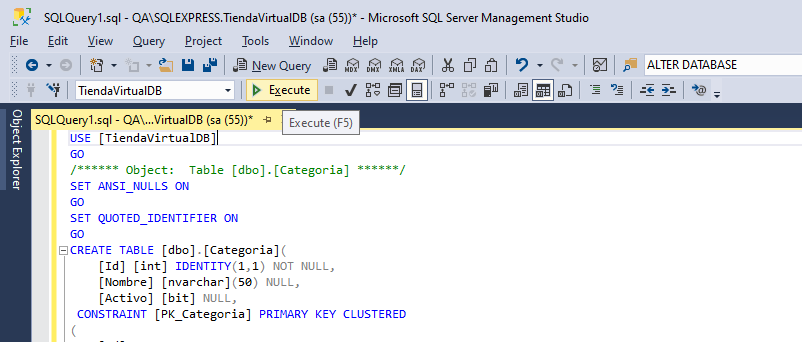
REFERENCES [dbo].[Marca] ([Id])

GO

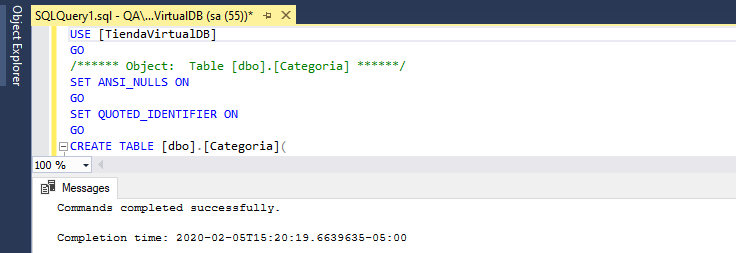
ALTER TABLE [dbo].[Producto] CHECK CONSTRAINT [FK\_Producto\_Marca]

GO

1. Para ejecutar el script presionamos en la opción de menú ***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:

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

GO

INSERT [dbo].[Categoria] ([Id], [Nombre], [Activo]) VALUES (1, N'Zapatillas', 1)

GO

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

GO

INSERT [dbo].[Categoria] ([Id], [Nombre], [Activo]) VALUES (3, N'Buzos', 0)

GO

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

GO

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

GO

INSERT [dbo].[Marca] ([Id], [Nombre]) VALUES (1, N'Asics')

GO

INSERT [dbo].[Marca] ([Id], [Nombre]) VALUES (2, N'Umbro')

GO

INSERT [dbo].[Marca] ([Id], [Nombre]) VALUES (3, N'Adidas')

GO

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

GO

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

GO

INSERT [dbo].[Producto] ([Id], [IdCategoria], [IdMarca], [Nombre], [Descripcion], [Precio], [Url], [Destacado], [Activo]) VALUES (1, 1, 1, N'GEL-NIMBUS 19', NULL, CAST(256.50 AS Decimal(10, 2)), N'https://images.freeimages.com/images/premium/previews/2444/24444520-stylized-photo-camera.jpg', 1, 1)

GO

INSERT [dbo].[Producto] ([Id], [IdCategoria], [IdMarca], [Nombre], [Descripcion], [Precio], [Url], [Destacado], [Activo]) VALUES (2, 1, 3, N'ZAPATILLA SOLAR GLIDE 19', NULL, CAST(449.00 AS Decimal(10, 2)), N'https://images.freeimages.com/images/premium/previews/2444/24444520-stylized-photo-camera.jpg', 0, 1)

GO

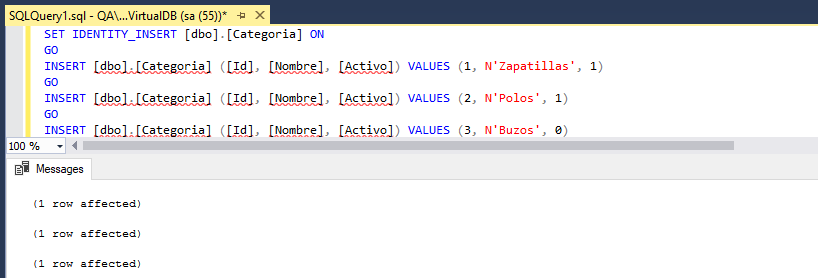
INSERT [dbo].[Producto] ([Id], [IdCategoria], [IdMarca], [Nombre], [Descripcion], [Precio], [Url], [Destacado], [Activo]) VALUES (3, 2, 3, N'POLO ID ALLOVER PRINT', NULL, CAST(109.00 AS Decimal(10, 2)), N'https://images.freeimages.com/images/premium/previews/2444/24444520-stylized-photo-camera.jpg', 1, 1)

GO

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

GO

1. Ejecutamos el script y verificamos que 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 qué Productos tenemos registrado en la BD ejecutamos:

SELECT P.Id, P.Nombre AS 'Producto', P.Precio, M.Nombre AS 'Marca', C.Nombre AS 'Categoria',

IIF(P.Destacado > 0, 'SI', 'NO') AS 'Destacado',

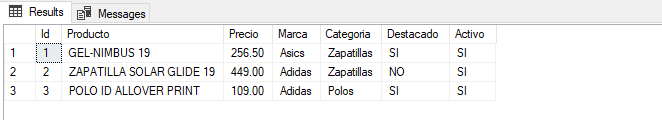
IIF(P.Activo > 0, 'SI', 'NO') AS 'Activo'

FROM Producto P

INNER JOIN Marca M ON P.IdMarca = M.Id

INNER JOIN Categoria C ON P.IdCategoria = C.Id

1. Debemos obtener:



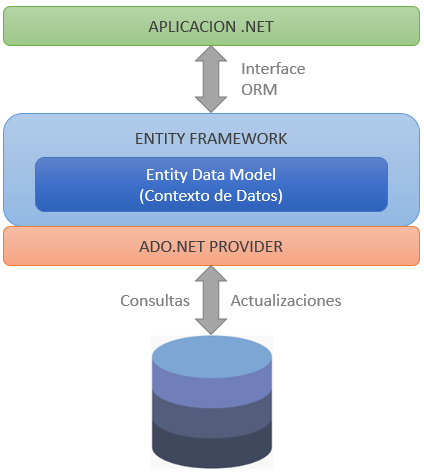
**Subtema 1.2:**

ENTITY FRAMEWORK CORE

Entity Framework (EF) Core es una versión ligera, extensible, de código abierto y multiplataforma de la popular tecnología de acceso a datos Entity Framework.

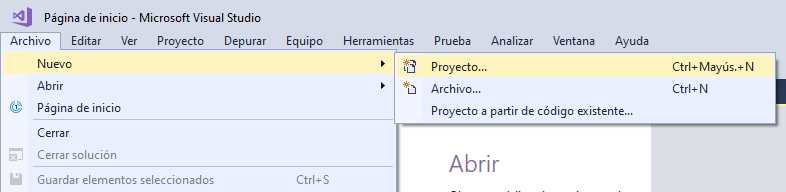
EF Core puede servir como un mapeador relacional de objetos (ORM), lo que permite a los desarrolladores de .NET trabajar con una base de datos mediante objetos .NET y eliminar la mayoría del código de acceso a los datos que normalmente deben escribir.

EF Core es compatible con muchos motores de base de datos como SQL Server, Oracle, PostgreSql, etc.



Paso 2. Crear un proyecto ASP.NET Core MVC en Visual Studio.

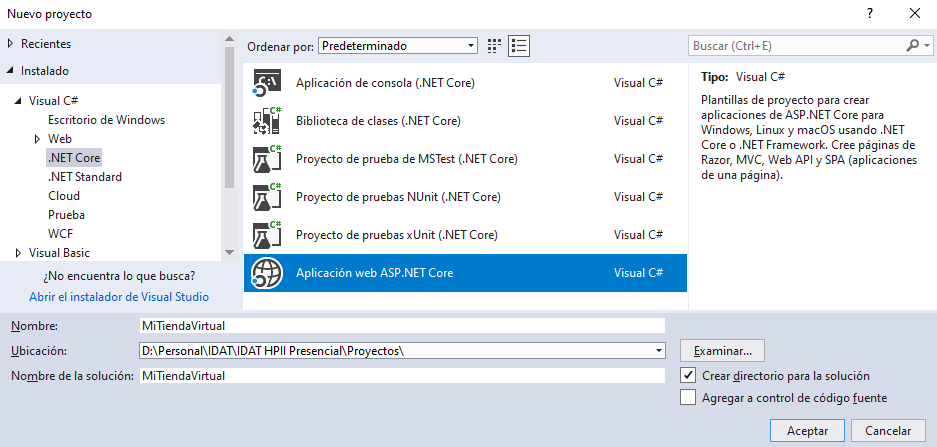
1. Abrimos Visual Studio.
2. Seleccionamos la opción: Archivo > Nuevo > ***Proyecto***…



1. En el formulario seleccionamos el tipo de proyecto:

Visual C# > .NET Core > ***Aplicación web ASP.NET Core***

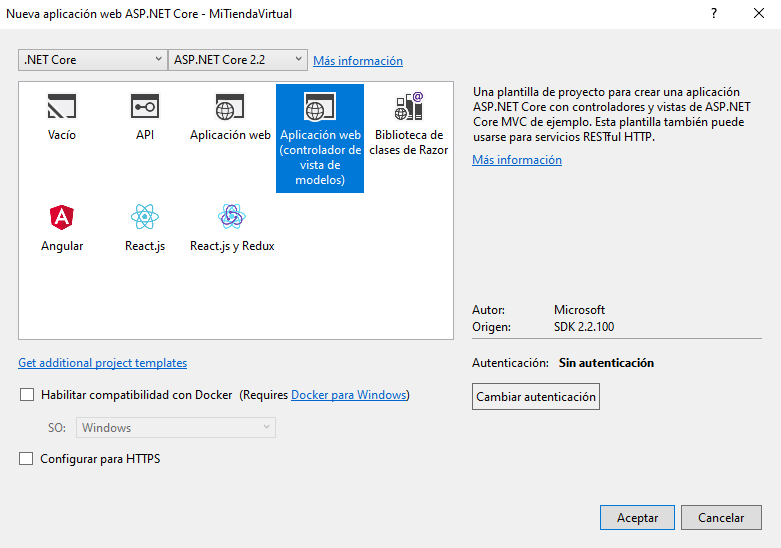
1. Le ponemos el nombre ***MiTiendaVirtual*** y presionamos en el botón ***Aceptar***.



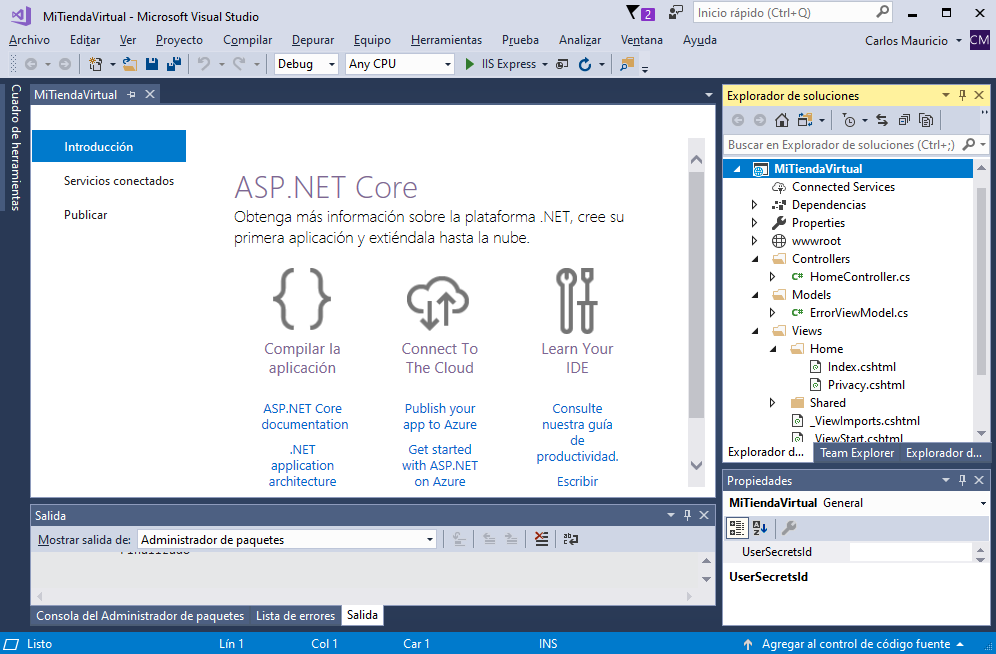
1. En el nuevo formulario seleccionamos la opción: ***Aplicación web (controlador de vista de modelos).***

Tener cuidado en la versión del .NET Core, debemos seleccionar la opción: ***ASP.NET Core 2.2.***

No debe estar seleccionado ninguno de los dos CheckBox.

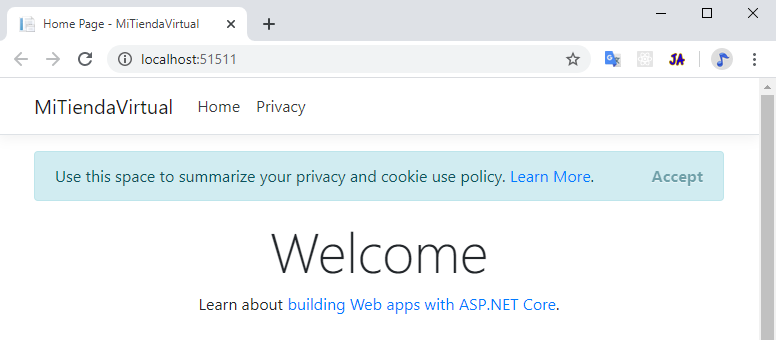


1. Presionamos en el botón Aceptar.
2. Verificamos la estructura del proyecto (MVC).



1. Ejecutamos la aplicación, para ellos seleccionamos la opción del menú: IIS Express.

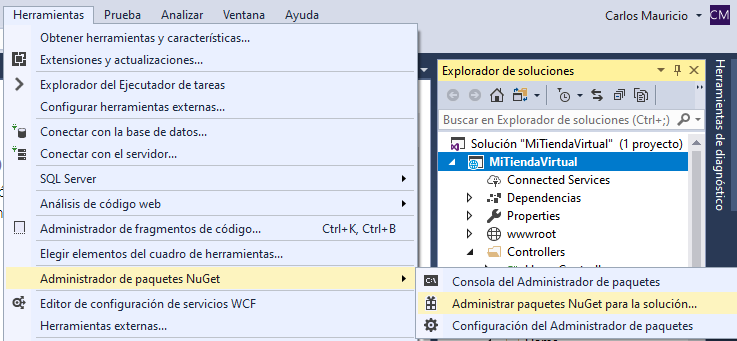




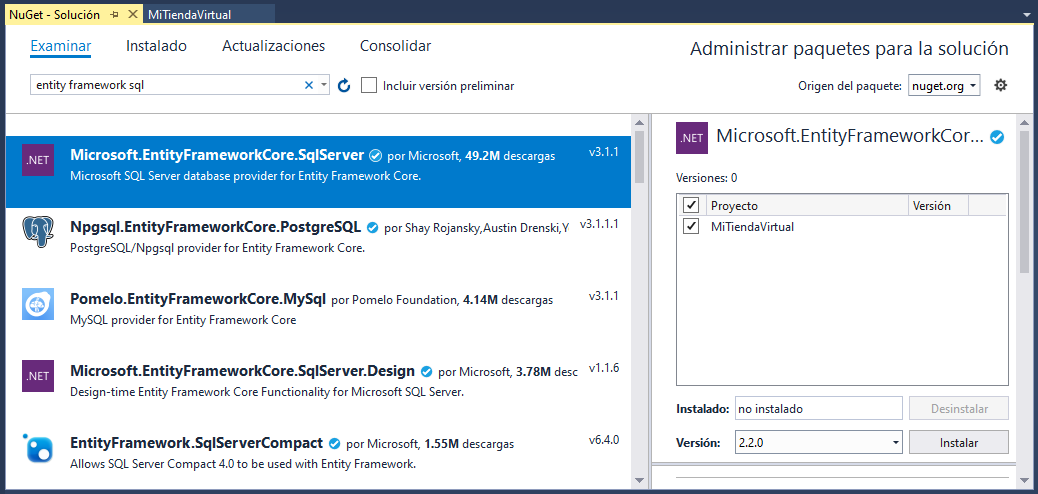
1. Para dejar de ejecutar la aplicación cerramos el explorador web.

Paso 3. Crear el Modelo de la aplicación a partir de la BD.

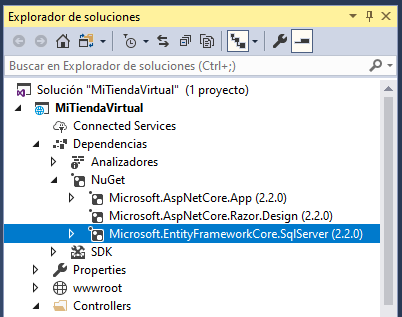
1. Antes de crear las entidades (clases) del Modelo debemos agregar a nuestra aplicación las librerías de Entity Framework Core. Seleccionamos la opción: ***Administrar paquetes NuGet para la solución…***



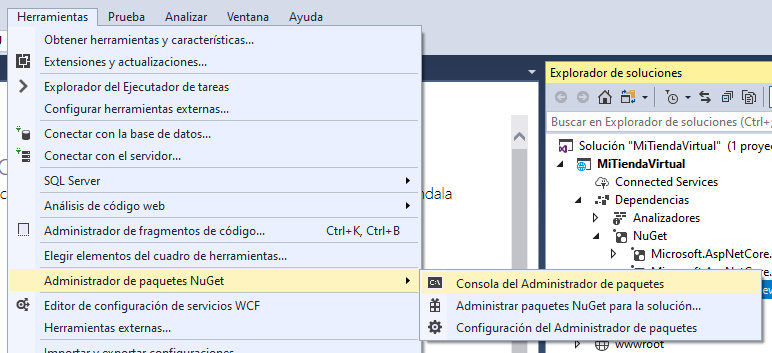
1. Dentro de la pestaña Examinar, en el cuadro de búsqueda ingresamos el texto: ***entity framework sql.***
2. De la lista resultante seleccionamos: ***Microsoft.EntityFrameworkCore.SqlServer.***
3. Hacemos ***check*** en el nombre de nuestro proyecto ***MiTiendaVirtual***.
4. Seleccionamos la versión ***2.2.0***
5. Presionamos en el botón ***Instalar***.

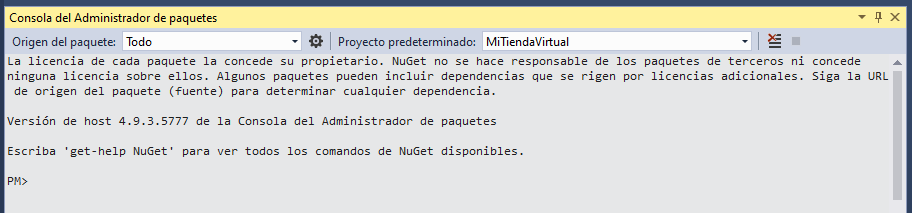


1. Verificamos el paquete instalado.



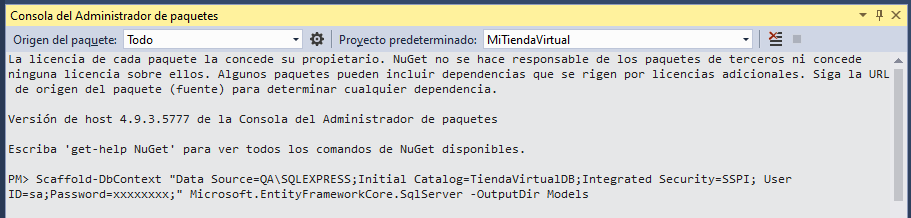
1. Ahora si vamos a generar el Modelo y el Contexto de Datos de nuestra aplicación, seleccionamos la opción: ***Consola del Administrador de paquetes***



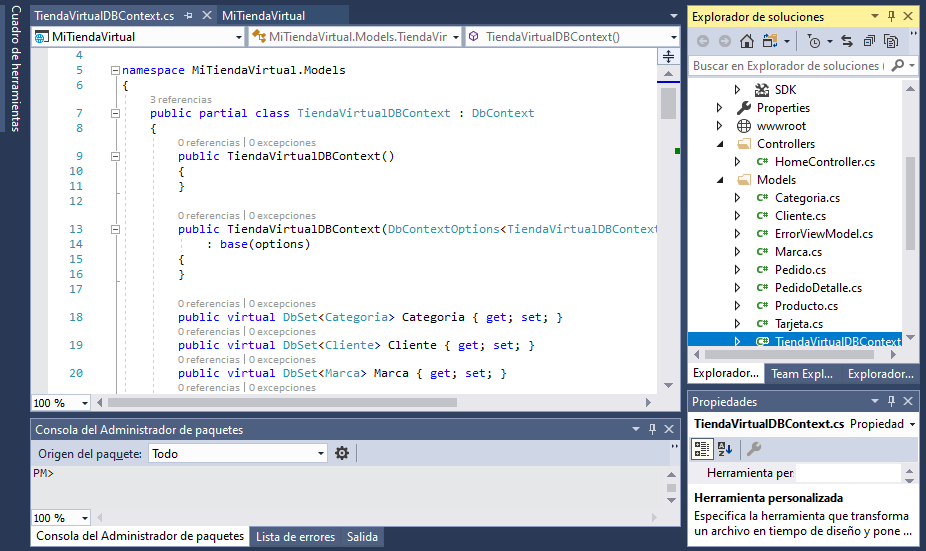


1. Ingresamos el siguiente comando:

Scaffold-DbContext "Data Source=QA\SQLEXPRESS;Initial Catalog=TiendaVirtualDB;Integrated Security=SSPI; User ID=sa;Password=\*\*\*\*\*\*\*\*;" Microsoft.EntityFrameworkCore.SqlServer -OutputDir Models

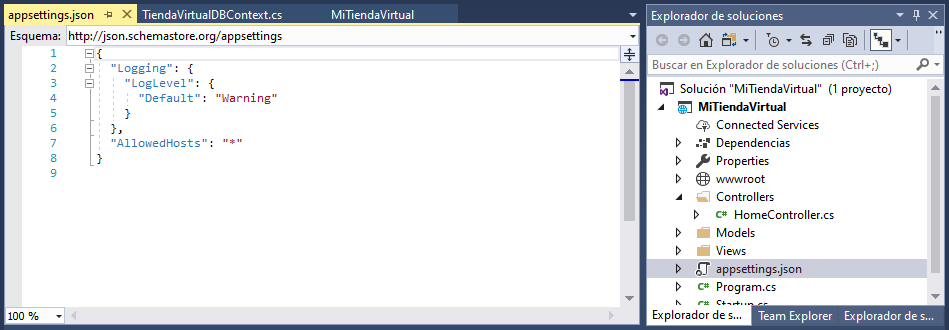


1. Debemos acondicionar la cadena de conexión de acuerdo a la configuración de la BD en SQL Server.
2. Presionamos la tecla Enter para ejecutar el comando.
3. Verificamos que se han creado las entidades (clases) del Modelo y su correspondiente Contexto de Datos.



Paso 4. Configurar la cadena de conexión a la BD.

1. Para agregar una cadena de conexión abrimos el archivo ***appsettings.json***.



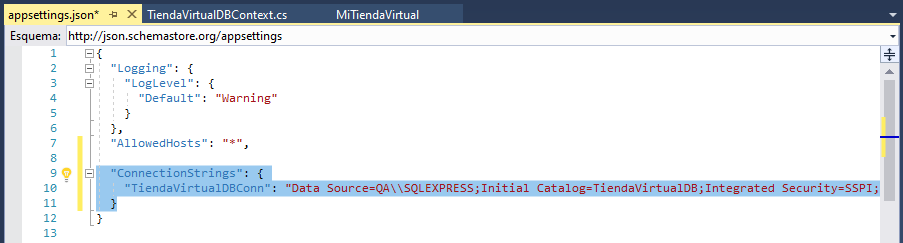
1. Agregamos la siguiente línea de código:

"ConnectionStrings": {

"TiendaVirtualDBConn": "Data Source=QA\\SQLEXPRESS;Initial Catalog=TiendaVirtualDB;Integrated Security=SSPI; User ID=sa;Password=\*\*\*\*\*\*\*"

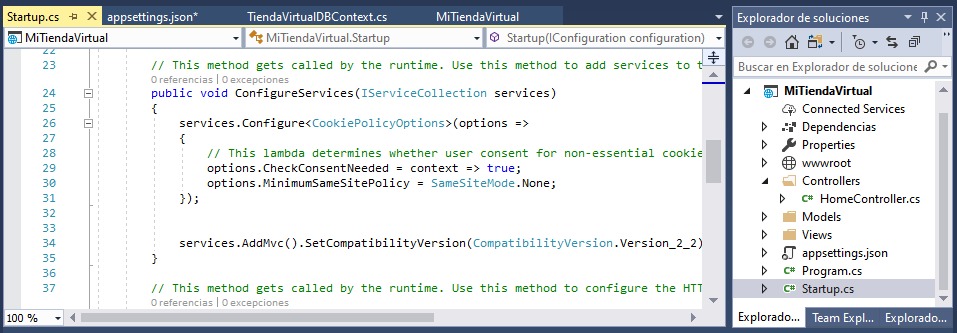
}

Recordar que se debe acondicionar la cadena de conexión de acuerdo a la configuración de la BD en SQL Server.



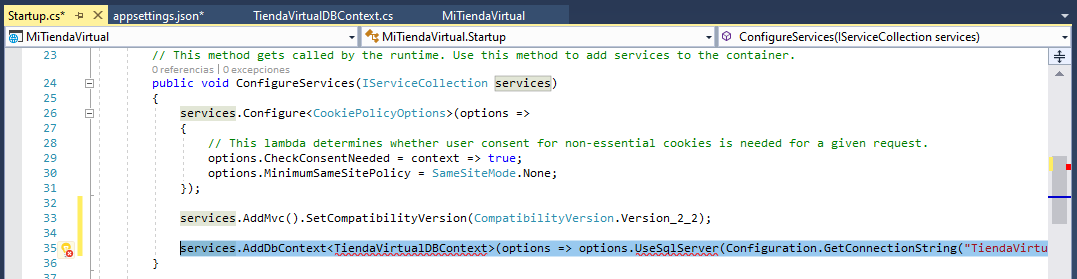
1. Observar que el nombre de la cadena de conexión es ***TiendaVirtualDBConn.***
2. Finalmente, para optimizar el acceso a la BD con Entity Framework Core vamos a registrar el Contexto de Datos como un Servicio dentro de nuestra aplicación (esto aplica el patrón de diseño denominado Inyección de Dependencias).

Abrimos el archivo ***Startup.cs***



1. Agregamos la siguiente línea de código:

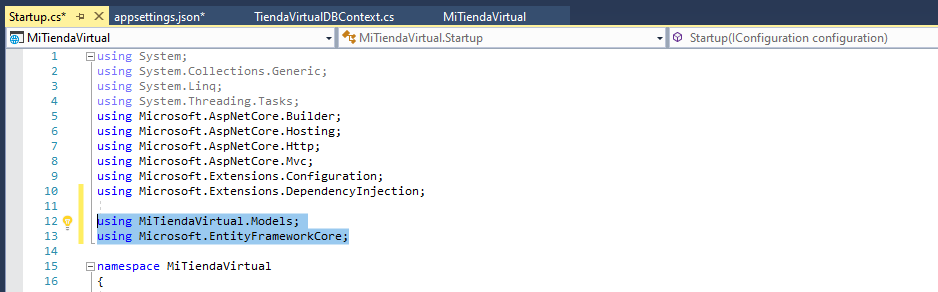
services.AddDbContext<TiendaVirtualDBContext>(options => options.UseSqlServer(Configuration.GetConnectionString("TiendaVirtualDBConn")));



1. Observamos que se muestran errores, para corregirlos debemos agregar en la parte superior de este mismo archivo las referencias a dos componentes de la aplicación:

using MiTiendaVirtual.Models;

using Microsoft.EntityFrameworkCore;



1. Ejecutar la aplicación para verificar que no existen errores.

CONCLUSIONES DE LA EXPERIENCIA

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**ACTIVIDAD VIRTUAL:**

Ingresa a la plataforma virtual, luego realiza las siguientes actividades propuestas:

1. CUESTIONARIO TÉCNICO
   * 1. Crear un script SQL que permita agregar datos a las tablas: Cliente, Tarjeta, Pedido y PedidoDetalle. No menos de cinco registros por tabla.
     2. Crear un script SQL que permita consultar los pedidos existentes en la BD, debe tener las siguientes columnas:

* Id del Pedido
* Fecha (sin hora) del Pedido
* Estado del Pedido
* Monto total del Pedido
* Nombres y apellidos del Cliente
* Marca de la Tarjeta
* Número de la Tarjeta

Ordenar ascendentemente por Nombre del Cliente y luego por Fecha del Pedido.

* + 1. Crear un script SQL que permita consultar los productos incluidos en un pedido, debe tener las siguientes columnas:
* Id del Pedido
* Nombre del Producto
* Cantidad
* Precio unitario
* Sub total

Ordenar ascendentemente por Nombre del Producto.

* + 1. Crear un procedimiento almacenado por cada uno de los dos últimos scripts de consulta creados anteriormente, indicar cuál de ellos debe recibir un parámetro.
    2. Investigar el procedimiento correcto a seguir para actualizar el Modelo si se agrega una nueva columna a alguna de las tablas, por ejemplo, si se va agregar la columna Activo (de tipo bit) a la tabla Marca.

1. CONCLUSIONES DE LA EXPERIENCIA

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_