# EF CORE DATABASE FIRST

1. Crear proyecto “ASP.NET CORE Web API” con .net core 5.0
2. Seleccionar el proyecto Web API e instalar paquetes nuget
   1. Microsoft.EntityFrameworkCore versión 5.0.17 (Esto porque el proyecto se creó con .net core 5.0)
   2. Microsoft.EntityFrameworkCore.Tools versión 5.0.17 (Esto porque el proyecto se creó con .net core 5.0)
   3. Microsoft.EntityFrameworkCore.SQLServer versión 5.0.17 (Esto porque el proyecto se creó con .net core 5.0)
   4. Microsoft.EntityFrameworkCore.Design versión 5.0.17 (Esto porque el proyecto se creó con .net core 5.0)
   5. Microsoft.Data.SqlClient
3. Crea la migración
   1. Ingresar a la URL <https://www.entityframeworktutorial.net/efcore/entity-framework-core.aspx>. Seleccionar EF Core – Existing DataBase
   2. Seleccionar el texto “Scaffold-DbContext "Server=.\SQLExpress;Database=SchoolDB;Trusted\_Connection=True;" Microsoft.EntityFrameworkCore.SqlServer -OutputDir Models”
   3. Actualizar el punto b con el nombre del servidor y la base de datos

“Scaffold-DbContext "Server=CESARIOS-YOGA;Database=Northwind;Trusted\_Connection=True;" Microsoft.EntityFrameworkCore.SqlServer -OutputDir Models -Force”

|  |  |
| --- | --- |
|  |  |

* 1. En visual studio ir a Tools-> Nuget Package Manager -> Package Manager Console
  2. Copiar y pegar el texto del punto c.
  3. Eliminar el método “OnConfiguring” que se creó en la clase “NorthwindContext.cs”

protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)

{

if (!optionsBuilder.IsConfigured)

{

#warning To protect potentially sensitive information in your connection string, you should move it out of source code. You can avoid scaffolding the connection string by using the Name= syntax to read it from configuration - see https://go.microsoft.com/fwlink/?linkid=2131148. For more guidance on storing connection strings, see http://go.microsoft.com/fwlink/?LinkId=723263.

optionsBuilder.UseSqlServer("Server=CESARIOS-YOGA;Database=Northwind;Trusted\_Connection=True;");

}

}

* 1. Configurar la cadena de conexión en el archivo “appsettings.json”

{

"Logging": {

"LogLevel": {

"Default": "Information",

"Microsoft": "Warning",

"Microsoft.Hosting.Lifetime": "Information"

}

},

"AllowedHosts": "\*",

"ConnectionStrings": {

"ConexionSQLServer": "server=CESARIOS-YOGA;database= Northwind;Trusted\_Connection=true;MultipleActiveResultSets=true;"

}

}

* 1. En la clase ‘NorthwindContext.cs’ eliminar el constructor que está vacio

public NorthwindContext()

{

}

* 1. Modificar el archivo ‘Startup.cs’ agregando el contexto para poder utilizarlo en inyección de dependencias. Se agrega en el contenedor de servicio el contexto “NorthwindContext”

public void ConfigureServices(IServiceCollection services)

{

services.AddControllers();

services.AddSwaggerGen(c =>

{

c.SwaggerDoc("v1", new OpenApiInfo { Title = "EFCoreDBFirst", Version = "v1" });

});

services.AddDbContext<NorthwindContext>(options=>options.UseSqlServer(Configuration.GetConnectionString("ConexionSQLServer")));

}

* 1. Crear controlador API en blanco con nombre por ejemplo ‘CustomerController.cs’ para manejar la entidad ‘Customer’ de la BD.