

13. Contexto de base de datos

La clase principal que coordina la funcionalidad de Entity Framework para un modelo de datos dado es la clase de contexto de la base de datos. Esta clase se crea derivando de la clase `Microsoft.EntityFrameworkCore.DbContext`. En su código se especifican que entidades se incluyen en el modelo de datos. También puede personalizar ciertos comportamientos de Entity Framework.

En este proyecto, la clase se nombra `SistemaContexto` y se creó de manera automática en la carpeta Data al momento de crear el primer controlador llamado `CategoriaController` (recordemos que el controlador también creó las vistas).

En la clase *SistemaContexto.cs*:

Por ahora solo tenemos un conjunto de entidades o tabla (clase) entonces solo incluimos esta clase en el contexto, más adelante a medida que agreguemos más tablas iremos actualizando nuestro contexto de base de datos.

Este código crea una propiedad `DbSet` para cada conjunto de entidades es decir para una tabla.

En la terminología de Entity Framework, un conjunto de entidades corresponde típicamente a una tabla de base de datos y una entidad corresponde a una fila en la tabla.

Registrar el contexto con la inyección de dependencia

ASP.NET Core implementa la inyección de dependencia de forma predeterminada. Los servicios (como el contexto de la base de datos EF) se registran con la inyección de dependencia durante el inicio de la aplicación. Los componentes que requieren estos servicios (como los controladores MVC) se proporcionan a través de parámetros de constructor.

El contexto de la base de datos `SistemaContexto` se registró como un servicio en *Startup.cs*.

```
public void ConfigureServices(IServiceCollection services)
{
    // Add framework services.
    var connection = Configuration.GetConnectionString("DefaultConnection");
    services.AddDbContext<SistemaContexto>(options =>
        options.UseSqlServer(connection));
}
```

```
}
```

El nombre de la cadena de conexión se pasa al contexto llamando a un método en un objeto `DbContextOptionsBuilder`. Para el desarrollo local, EL Sistema de configuración [ASP.NET Core](https://docs.microsoft.com/en-us/aspnet/core/mvc/models/working-with-ef?view=aspnetcore-2.2) lee la cadena de conexión del archivo *appsettings.json*.

Cadena de conexión

Revise la cadena de conexión generada automáticamente el archivo *appsettings.json* y agregue una cadena de conexión como se muestra en el ejemplo siguiente:

```
{
  "ConnectionStrings": {
    "DefaultConnection":
    "Server=(localdb)\\mssqllocaldb;Database=sistema;Trusted_Connection=True;MultipleActiveRe
    sultSets=true"
  },
  "Logging": {
    "IncludeScopes": false,
    "LogLevel": {
      "Default": "Warning"
    }
  }
}
```

Para ver más opciones de configuración de la cadena de conexión puedes revisar la siguiente url:

<https://msdn.microsoft.com/en-us/library/jj653752.aspx>

SQL Server Express LocalDB

La cadena de conexión especifica una base de datos SQL Server LocalDB. LocalDB es una versión ligera del motor de base de datos de SQL Server Express y está destinado al desarrollo de aplicaciones, no al uso de producción. LocalDB se inicia a petición y se ejecuta en modo de usuario, por lo que no hay una configuración compleja. De forma predeterminada, LocalDB crea archivos de base de datos .mdf en el directorio `C:/Users/<user>`