**Instalar o Entity Framework 6**

1. **ferramentas** menu, escolha **Gerenciador de pacotes NuGet**, em seguida, escolha **Package Manager Console**.
2. No **Package Manager Console** janela, digite o seguinte comando:

Install-Package EntityFramework

Esta etapa é uma das poucas etapas que este tutorial tem a fazer manualmente, mas que poderia ter sido feito automaticamente pelo recurso de scaffolding do ASP.NET MVC. Você está fazendo-os manualmente para que você possa ver as etapas necessárias para usar o Entity Framework (EF). Você vai usar scaffolding mais tarde para criar o controlador MVC e exibições. Uma alternativa é permitir que o scaffolding automaticamente instala o pacote NuGet do EF, criar a classe de contexto do banco de dados e criar a cadeia de caracteres de conexão. Quando você estiver pronto para fazer isso dessa maneira, tudo o que você precisa fazer é ignorar essas etapas e o scaffolding seu controlador MVC, depois de criar as classes de entidade.

## Criar o contexto de banco de dados

A classe principal que coordena a funcionalidade do Entity Framework para determinado modelo de dados é o contexto de banco de dados classe. Você cria essa classe derivando de [System.Data.Entity.DbContext](https://msdn.microsoft.com/library/system.data.entity.dbcontext(v=vs.113).aspx) classe. No seu código, você especifica quais entidades são incluídas no modelo de dados. Também personalize o comportamento específico do Entity Framework. Neste projeto, a classe é chamada SchoolContext.

* Para criar uma pasta no projeto ContosoUniversity, clique com botão direito no projeto no **Gerenciador de soluções** e clique em **Add**e, em seguida, clique em**nova pasta**. Nomeie a nova pasta DAL (para a camada de acesso a dados). Nessa pasta, crie um novo arquivo de classe chamado SchoolContext.cse substitua o código de modelo pelo código a seguir:

using ContosoUniversity.Models;

using System.Data.Entity;

using System.Data.Entity.ModelConfiguration.Conventions;

namespace ContosoUniversity.DAL

{

public class SchoolContext : DbContext

{

public SchoolContext() : base("SchoolContext")

{

}

public DbSet<Student> Students { get; set; }

public DbSet<Enrollment> Enrollments { get; set; }

public DbSet<Course> Courses { get; set; }

protected override void OnModelCreating(DbModelBuilder modelBuilder)

{

modelBuilder.Conventions.Remove<PluralizingTableNameConvention>();

}

}

}

### Especificar conjuntos de entidades

Esse código cria uma [DbSet](https://msdn.microsoft.com/library/system.data.entity.dbset(v=vs.113).aspx) propriedade para cada conjunto de entidades. Na terminologia do Entity Framework, um conjunto de entidades normalmente corresponde a uma tabela de banco de dados e um entidade corresponde a uma linha na tabela.

**Observação**

Você pode omitir as DbSet<Enrollment> e DbSet<Course> instruções e ele funcionaria da mesma. Entity Framework inclui-los implicitamente porque o Student referências de entidade a Enrollment entidade e o Enrollmentreferências de entidade a Course entidade.

### Especifique a cadeia de conexão

O nome da cadeia de conexão (que você irá adicionar posteriormente no arquivo Web. config) é passado para o construtor.

C#Copiar

public SchoolContext() : base("SchoolContext")

{

}

Você também pode transmitir a cadeia de conexão em si em vez do nome de um que é armazenado no arquivo Web. config. Para obter mais informações sobre opções para especificar o banco de dados a serem usadas, consulte [cadeias de caracteres de Conexão e modelos](https://docs.microsoft.com/pt-br/ef/ef6/fundamentals/configuring/connection-strings).

Se você não especificar uma cadeia de caracteres de conexão ou o nome de um explicitamente, o Entity Framework pressupõe que o nome de cadeia de caracteres de conexão é o mesmo que o nome da classe. O nome de cadeia de caracteres de conexão padrão neste exemplo, em seguida, seria SchoolContext, o mesmo que o que você especificar explicitamente.

### Especificar nomes singulares de tabela

O modelBuilder.Conventions.Remove instrução em de [OnModelCreating](https://msdn.microsoft.com/library/system.data.entity.dbcontext.onmodelcreating(v=vs.103).aspx) método impede que os nomes de tabela que está sendo pluralizados. Se você não fizer isso, as tabelas geradas no banco de dados seriam nomeadas Students, Courses, e Enrollments. Em vez disso, os nomes de tabela serão Student, Course, e Enrollment. Os desenvolvedores não concordam sobre se os nomes de tabela devem ser pluralizados ou não. Este tutorial usa a forma singular, mas o ponto importante é que você pode selecionar qualquer que seja o formulário que você preferir ao incluir ou omitir esta linha de código.

No cenário do nosso projeto, iremos criar uma classe com as configurações preparadas para banco de dados, utilizando o conceito de “Data Annotations”.

using System;

using System.ComponentModel.DataAnnotations;

using System.ComponentModel.DataAnnotations.Schema;

namespace ControleFinanceiro.Models

{

[Table("Cliente")]

public class Cliente

{

[Key]

public int ClienteId { get; set; }

[MaxLength(18)]

public string CPF { get; set; }

[MaxLength(50)]

public string Nome { get; set; }

[MaxLength(15)]

public string Telefone { get; set; }

[MaxLength(100)]

public string Email { get; set; }

public DateTime Nascimento { get; set; }

public bool Ativo { get; set; }

}

}

**MIGRATIONS**

Referência: <http://www.entityframeworktutorial.net/code-first/code-based-migration-in-code-first.aspx>

**Enable-Migrations**

Habilita a migração no projeto criando uma classe Configuration.

**Add-Migration <nome da migration>**

Cria uma nova classe de migração conforme o nome especificado com os métodos Up () e Down ().

Exemplo: Add-Migration InicioDoProjeto

**Update-Database**

Executa o último arquivo de migração criado pelo comando Add-Migration e aplica alterações ao esquema do banco de dados.

**Update-Database -script**

Exibe o script de migração baseado no último arquivo de migração criado.

Referências

<https://docs.microsoft.com/pt-br/aspnet/mvc/overview/getting-started/getting-started-with-ef-using-mvc/creating-an-entity-framework-data-model-for-an-asp-net-mvc-application>

<https://www.entityframeworktutorial.net/EntityFramework6/introduction.aspx>