using System;

using System.Collections.Generic;

using System.ComponentModel.DataAnnotations.Schema;

using System.Data.Entity;

using System.Data.Entity.ModelConfiguration.Conventions;

using System.Linq;

using System.Web;

namespace EfCodeFirst.Models

{

public class ClassContext: DbContext

{

public ClassContext()

: base("cadenaconexion1")

{

}

public DbSet<BlogPost> BlogPosts { get; set; }

public DbSet<Comentario> Comentarios { get; set; }

//L61 para utilizar API FLUENTE hay que facer un override ó metodo OnModelCreating e mostrara o parametro modelBuilder

protected override void OnModelCreating(DbModelBuilder modelBuilder)

{

## //Definimos clave primaria

modelBuilder.Entity<BlogPost>().HasKey(x => x.Id);

## //Definimos clave primaria Compuesta

modelBuilder.Entity<Comentario>().HasKey(x => new { x.Autor,x.Contenido});

## //Definimos que o valor da clave primaria nn veña por defecto e o teñamos que asignar nos

modelBuilder.Entity<Comentario>().Property(x => x.Id).HasDatabaseGeneratedOption(DatabaseGeneratedOption.None);

## //Campo con longitud fija e maxima

modelBuilder.Entity<BlogPost>().Property(x => x.Titulo).IsFixedLength().HasMaxLength(11);

/\*Podriase poñer en duas filas:

modelBuilder.Entity<BlogPost>().Property(x => x.Titulo).IsFixedLength();

modelBuilder.Entity<BlogPost>().Property(x => x.Titulo).HasMaxLength(11);

\*/

## //Todas as propiedades int que empecen por codigo seran claves primarias

modelBuilder.Properties<int>().Where(x => x.Name.StartsWith("Codigo")).Configure(x => x.IsKey());

//logitud dun campo

modelBuilder.Entity<BlogPost>().Property(x => x.Contenido).HasMaxLength(150);

## //Propiedade requerida

modelBuilder.Entity<BlogPost>().Property(x => x.Edad).IsRequired();

## //Propiedad no se mapea un una columna

modelBuilder.Entity<BlogPost>().Ignore(x => x.ConfirmarEmail);

## //Definimos o nombre que tendrá unha columna independientemente de como se chame o campo

modelBuilder.Entity<BlogPost>().Property(x => x.CampoSecreto).HasColumnName("Campo\_Secreto");

## //Definir o nome que terá unha tabla

modelBuilder.Entity<Comentario>().ToTable("Comentarios");

.

## //CONVENCIONES

### //No pluraliza nombres de tablas

modelBuilder.Conventions.Remove<PluralizingTableNameConvention>();

### //No mapea decimales a numeric(18,2)

modelBuilder.Conventions.Remove<DecimalPropertyConvention>();

### //Los decimales serán decima(16,2)

modelBuilder.Properties<decimal>().Configure(x => x.HasColumnType("decimal").HasPrecision(16, 2);

//ejecutamos add-migration e update-database e cambiaranse as configuracions

base.OnModelCreating(modelBuilder);

}

}

}