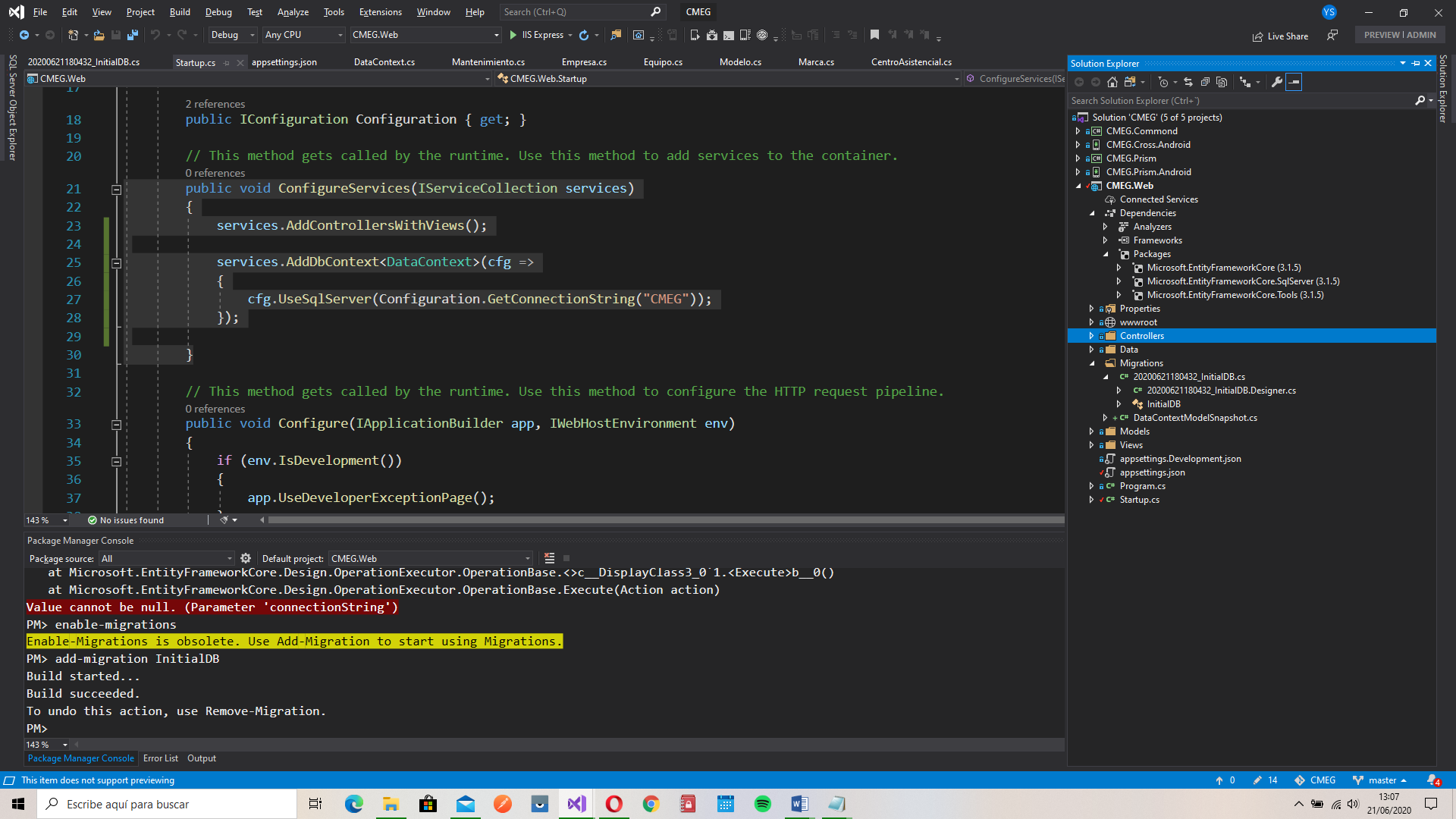
ERRORES AL TENER INSTALADO ASP NET CORE 3.1

* UseSqlServer, sale error porque esta instalado el net core 3.1 por lo tanto se debe de instalar el nuget

public void ConfigureServices(IServiceCollection services)

{

services.AddControllersWithViews();

services.AddDbContext<DataContext>(cfg =>

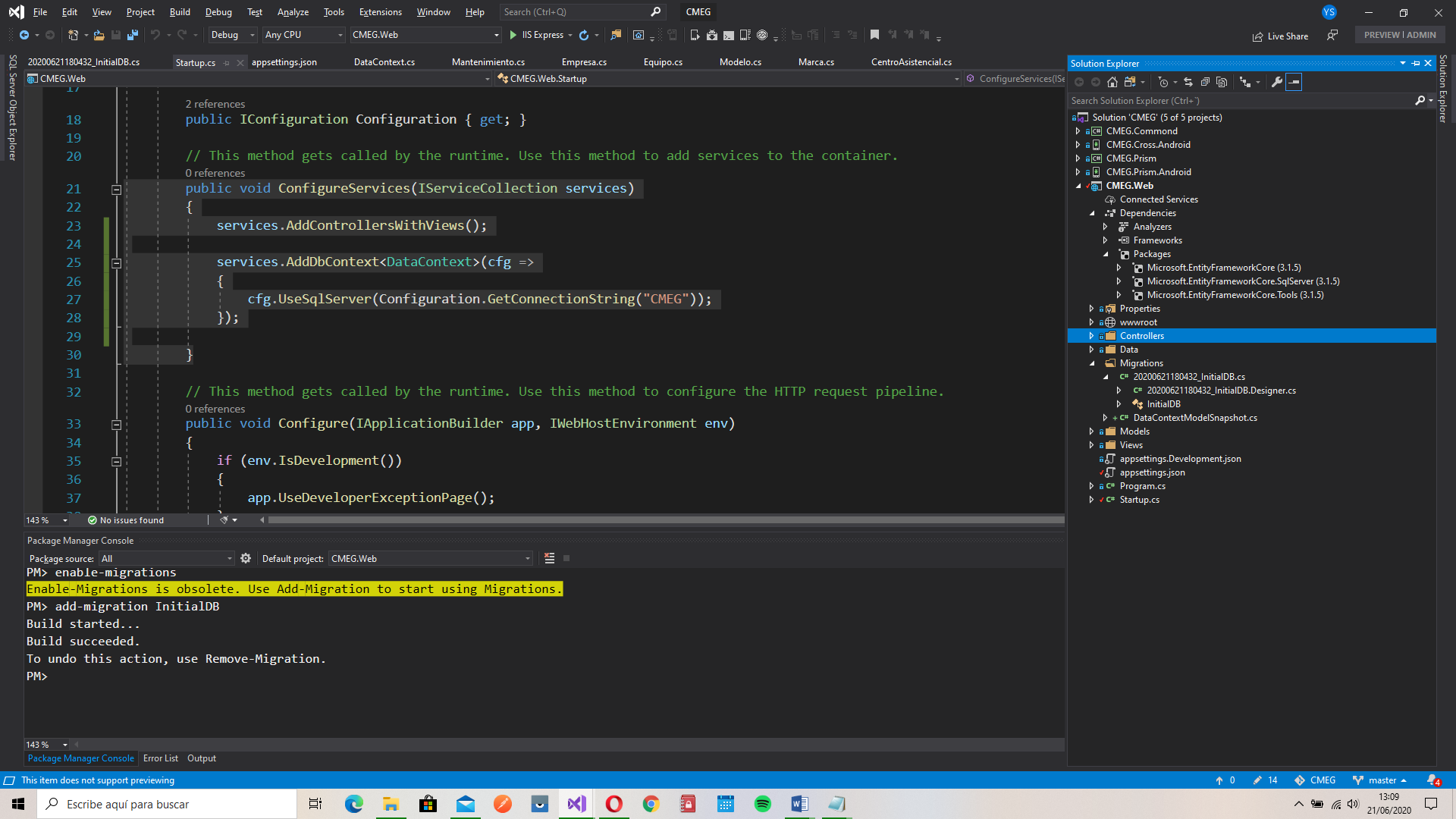
{

cfg.UseSqlServer(Configuration.GetConnectionString("CMEG"));

});

}

* Al querer crear migraciones con el comando add-migration de la consola NugetPackcageManager me sale el error que no reconoce dicho comando, para ello hay que instalar el paquete



* Para utilizar los glyphicon hay que agregar en index las siguientes lìneas

<link rel="stylesheet"href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css" />

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>

<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>

* Llenar un combo

Primero

public interface IComboBoxHelper

{

IEnumerable<SelectListItem> GetComboEquipos();

}

Segundo

using CMEG.Web.Data;

using Microsoft.AspNetCore.Mvc.Rendering;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

namespace CMEG.Web.Helpers

{

public class ConboBoxHelper : IComboBoxHelper

{

private readonly DataContext \_context;

public ConboBoxHelper(DataContext context)

{

\_context = context;

}

public IEnumerable<SelectListItem> GetComboEquipos()

{

List<SelectListItem> list = \_context.Equipos.Select(e => new SelectListItem

{

Text = e.Nombre,

Value = $"{e.Id}"

})

.OrderBy(e => e.Text)

.ToList();

return list;

}

}

}

Tercero

public interface IconverterHelper

{

Task<Mantenimiento> ToMantenimientoEntityAsync(MantenimientoViewModel model, bool isNew);

MantenimientoViewModel ToMantenimientoViewModel(Mantenimiento entity);

}

Cuarto

public class ConverterHelper : IConverterHelper

{

public async Task<Mantenimiento> ToMantenimientoEntityAsync(MantenimientoViewModel model, bool isNew)

{

return new Mantenimiento

{

AñoEjecucion = model.AñoEjecucion,

DetalleMantenimientos = model.DetalleMantenimientos,

FechaProgramacion = model.FechaProgramacion,

Id = isNew ? 0 : model.Id,

NumeroMantenimiento = model.NumeroMantenimiento,

};

}

public MantenimientoViewModel ToMantenimientoViewModel(Mantenimiento entity) {

return new MantenimientoViewModel

{

AñoEjecucion = entity.AñoEjecucion,

DetalleMantenimientos = entity.DetalleMantenimientos,

FechaProgramacion = entity.FechaProgramacion,

Id = entity.Id,

NumeroMantenimiento = entity.NumeroMantenimiento,

Equipos = \_combo.GetComboEquipos()

};

}

}

Quinto

En la vista de Create

<div class="form-group">

<label **asp-for**="IdEquipo" class="control-label"></label>

<select **asp-for**="IdEquipo" **asp-items**="Model.Equipos" class="form-control"></select>

<span **asp-validation-for**="IdEquipo" class="text-danger"></span>

</div>

Sexto

En el controller/create

public IActionResult Create()

{

Equipo entity = \_context.Equipos.FirstOrDefault();

if (entity == null)

{

return NotFound();

}

MantenimientoViewModel model = new MantenimientoViewModel

{

//Equipo = entity,

IdEquipo = entity.Id,

Equipos = \_combo.GetComboEquipos()

};

return View(model);

}

[HttpPost]

[ValidateAntiForgeryToken]

public async Task<IActionResult> Create(MantenimientoViewModel model)

{

if (ModelState.IsValid)

{

Mantenimiento entity = await \_converter.ToMantenimientoEntityAsync(model, true);

\_context.Add(model);

await \_context.SaveChangesAsync();

return RedirectToAction(nameof(Index));

}

model.Equipos = \_combo.GetComboEquipos();

return View(model);

}

Sèptimo

Agregar un viewmodel con los campos necesarios que no afecten a la entidad

namespace CMEG.Web.Models

{

public class MantenimientoViewModel : Mantenimiento

{

[Required(ErrorMessage = "El campo {0} es obligatorio")]

[Display(Name = "Equipo")]

[Range(1, int.MaxValue, ErrorMessage = "Selecciona un Equipo")]

public int IdEquipo { get; set; }

public IEnumerable<SelectListItem> Equipos { get; set; }

}

}

* dsd