#region USINGS

using Grpc.Net.Client;

using Microsoft.EntityFrameworkCore;

using ModuleHub.DataAccess.Contexts;

using ModuleHub.Domain.Entities;

using ModuleHub.Domain.Entities.Common;

using ModuleHub.Domain.Utilities.Types;

using ModuleHub.Protos.DataSource;

using ModuleHub.Protos;

using System;

using System.ComponentModel.DataAnnotations.Schema;

using ModuleHub.Protos.CommunicationClient;

using ModuleHub.Contracts.Interfaces;

using ModuleHub.Contracts;

using MediatR;

using AutoMapper;

using ModuleHub.DataAccess.Repositories.Common;

using ModuleHub.DataAccess;

using static ModuleHub.Protos.DataSource.DataSource;

using ModuleHub.Protos.OPCNode;

using ModuleHub.Protos.ModbusNode;

#endregion

namespace ModuleHub.ConsoleApp

{

internal class Program

{

static async Task Main(string[] args)

{

// if (File.Exists("ModuleHub\_DB.sqlite"))

// File.Delete("ModuleHub\_DB.sqlite");

string connectionString = "Data Source = ModuleHub\_DB.sqlite";

// contexto creado para interactuar con la Base Datos

ApplicationContext applicationContext = new ApplicationContext(connectionString);

//comprobando exitencia de la base de datos

if (!applicationContext.Database.CanConnect())

{

// si no existe BD entonces se crea ahora a partir de la migracion realizada

applicationContext.Database.Migrate();

}

#region\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* INTERACCION CON CLIENTE \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

#region CONEXION y CONFIG

Console.Clear();

//Inicio del programa

Console.WriteLine(" ESTABLICHING CONECCTION....");

Console.WriteLine("");

Console.WriteLine(" Press Any KEY \_\_....");

Console.WriteLine("");

Console.WriteLine(" Proccessing information....");

Console.ReadKey();

Console.WriteLine("");

//Creando un canal y un cliente

Console.WriteLine(" Creating Channel and Client.....");

var httpHandler = new HttpClientHandler();

httpHandler.ServerCertificateCustomValidationCallback = HttpClientHandler.DangerousAcceptAnyServerCertificateValidator;

var channel = GrpcChannel.ForAddress("http://localhost:5198", new GrpcChannelOptions { HttpHandler = httpHandler });

//Error en la creación del canal, retorno

if (channel is null)

{

Console.WriteLine("");

Console.WriteLine(" !! CANNOT CONNECT ¡¡");

channel.Dispose();

return;

}

Console.ReadKey();

Console.WriteLine(" Create Channel SUCCESFULL.....");

Console.ReadKey();

Console.Clear();

var DataSourceClient = new ModuleHub.Protos.DataSource.DataSource.DataSourceClient(channel);

var CommunicationClientClient = new ModuleHub.Protos.CommunicationClient.CommunicationClient.CommunicationClientClient(channel);

var ModbusNodeClient = new ModuleHub.Protos.ModbusNode.ModbusNode.ModbusNodeClient(channel);

var OPCNodeClient = new ModuleHub.Protos.OPCNode.OPCNode.OPCNodeClient(channel);

#endregion

#region \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*+\*\*\* MENU CLIENT CONSOLE \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*+\*\*

int Try = 1;

string AccessPassword = "ModuleHub";

string Password;

Console.WriteLine(" \*\*\*\*\*\*\*\*\*\*\*\*\*\* CLIENT APPLICATION \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* ");

Console.WriteLine(" \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* MODULE HUB CONFIGURATION \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

Console.WriteLine("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

do

{

Console.WriteLine("");

Console.WriteLine("");

Console.WriteLine(" Enter Password: ");

Password = Console.ReadLine();

if (Password == AccessPassword)

{

Console.WriteLine("");

Console.WriteLine("");

Console.Clear();

Console.WriteLine(" \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* WELCOME \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

Console.WriteLine("");

Console.WriteLine("");

bool cicle = true;

string OptionSelect = "";

string DateTypeSelect = "";

while (cicle == true)

{

Console.WriteLine(" Menu \n" +

"---------------------------------\n" +

"---------------------------------\n\n " +

" 1- Create \n" +

" 2- Get \n" +

" 3- Update \n" +

" 4- Delete \n" +

" 5- Finish \n");

Console.WriteLine("\n\n\n Please, Enter the # Option Requerid..... ");

OptionSelect = Console.ReadLine();

Console.Clear();

if (OptionSelect == "1" || OptionSelect == "2" || OptionSelect == "3" || OptionSelect == "4")

{

bool cicle2 = true;

while (cicle2 == true)

{

Console.WriteLine(" Select Entity Type Requerid.... \n" +

"-------------------------------- \n\n" +

" 1- Data Source \n" +

" 2- Communication Client \n" +

" 3- OPC Node \n" +

" 4- MODBUS Node \n" +

" 5- Back \n");

DateTypeSelect = Console.ReadLine();

Console.Clear();

if (DateTypeSelect == "1" || DateTypeSelect == "2" || DateTypeSelect == "3" || DateTypeSelect == "4" || DateTypeSelect == "5")

{

cicle2 = false;

if (DateTypeSelect == "5")

OptionSelect = "";

}

}

}// Seleccion del tipo de dato

switch (OptionSelect)

{

case "1":

switch (DateTypeSelect)

{

case "1":

#region CREATING DATA\_SOURCE

Console.WriteLine("");

Console.WriteLine("");

Console.WriteLine("");

Console.WriteLine(" Press Any Key To Create The DATA SOURCE....");

Console.WriteLine("");

Console.WriteLine(" Proccessing....");

Console.ReadKey();

Console.WriteLine("");

var createResponse0 = DataSourceClient.CreateDataSource(new CreateDataSourceRequest()

{

Code = "DATA\_SOURCE\_1",

InputsCounter = 5,

OutputsCounter = 3

});

//Fallo en la creación del mensaje, retorno.

if (createResponse0 is null)

{

Console.WriteLine(" Cannot create Data Source.....");

channel.Dispose();

return;

}

//Mensaje creado

else

{

Console.WriteLine(" \*\*\*\*\*\*\*\* DATA\_SOURCE \*\*\*\*\*\*\*\* ");

Console.WriteLine(" ¡¡ SUCCESFULL CREATION !!");

Console.WriteLine("");

Console.WriteLine(""); Console.WriteLine(""); Console.WriteLine("");

}

#endregion

break;

case "2":

#region CREATING COMMUNICATION\_CLIENT

Console.WriteLine("");

Console.WriteLine("");

Console.WriteLine("");

Console.WriteLine(" Press Any Key To Create The COMMUNICATION CLIENT....");

Console.WriteLine("");

Console.WriteLine(" Proccessing....");

Console.ReadKey();

Console.WriteLine("");

var createResponse1 = CommunicationClientClient.CreateCommunicationClient(new CreateCommunicationClientRequest()

{

AddressIp = "102.34.55.86",

DataSource = createResponse0

});

//Fallo en la creación del mensaje, retorno.

if (createResponse1 is null)

{

Console.WriteLine(" Cannot create Communication Client.....");

channel.Dispose();

return;

}

//Mensaje creado

else

{

Console.WriteLine(" \*\*\*\*\*\*\*\* COMMUNICATION CLIENT \*\*\*\*\*\*\*\* ");

Console.WriteLine(" ¡¡ SUCCESFULL CREATION !!");

Console.WriteLine("");

Console.WriteLine(""); Console.WriteLine(""); Console.WriteLine("");

}

#endregion

break;

case "3":

#region CREATING OPC\_NODE

Console.WriteLine("");

Console.WriteLine("");

Console.WriteLine("");

Console.WriteLine(" Press Any Key To Create The OPC NODE....");

Console.WriteLine("");

Console.WriteLine(" Proccessing....");

Console.ReadKey();

Console.WriteLine("");

var createResponse2 = OPCNodeClient.CreateOPCNode(new CreateOPCNodeRequest()

{

AddressLabel = "OPC\_NODE\_1",

CommunicationClient = createResponse1

});

//Fallo en la creación del mensaje, retorno.

if (createResponse2 is null)

{

Console.WriteLine(" Cannot create OPC NODE.....");

channel.Dispose();

return;

}

//Mensaje creado

else

{

Console.WriteLine(" \*\*\*\*\*\*\*\* OPC\_NODE \*\*\*\*\*\*\*\* ");

Console.WriteLine(" ¡¡ SUCCESFULL CREATION !!");

Console.WriteLine("");

Console.WriteLine(""); Console.WriteLine(""); Console.WriteLine("");

}

#endregion

break;

case "4":

#region CREATING MODBUS\_NODE

Console.WriteLine("");

Console.WriteLine("");

Console.WriteLine("");

Console.WriteLine(" Press Any Key To Create The MODBUS NODE....");

Console.WriteLine("");

Console.WriteLine(" Proccessing....");

Console.ReadKey();

Console.WriteLine("");

var createResponse3 = ModbusNodeClient.CreateModbusNode(new CreateModbusNodeRequest()

{

Name = "MODBUS\_NODE\_1",

RecordSource = 6,

CommunicationClient = createResponse1

});

//Fallo en la creación del mensaje, retorno.

if (createResponse3 is null)

{

Console.WriteLine(" Cannot create MODBUS NODE.....");

channel.Dispose();

return;

}

//Mensaje creado

else

{

Console.WriteLine(" \*\*\*\*\*\*\*\* MODBUS\_NODE \*\*\*\*\*\*\*\* ");

Console.WriteLine(" ¡¡ SUCCESFULL CREATION !!");

Console.WriteLine("");

Console.WriteLine(""); Console.WriteLine(""); Console.WriteLine("");

}

#endregion

break;

default:

Console.WriteLine("Leaving");

break;

}

break;

case "2":

switch (DateTypeSelect)

{

case "1":

Console.WriteLine(" Select How Many Items Do You Requerid...\n");

Console.WriteLine("------------------------------------------\n\n");

Console.WriteLine(" 1- Get 1 Item By Id");

Console.WriteLine(" 2- Get All Items From DataBase\n\n\n");

switch (Console.ReadLine())

{

case "1":

Console.WriteLine("");

Console.WriteLine("");

Console.WriteLine($" Press Any Key To Get The DATA SOURCE With ID => [ {createResponse0.Id} ] ");

Console.WriteLine("");

Console.WriteLine(" Proccessing....");

Console.WriteLine("");

Console.ReadKey();

var getByIdResponse0 = DataSourceClient.GetDataSource(new GetRequest() { Id = createResponse0.Id.ToString() });

if (getByIdResponse0 is null)

{

Console.WriteLine(" Cannot get DATA SOURCE");

channel.Dispose();

return;

}

else

{

Console.WriteLine("");

}

Console.WriteLine(" \*\*\*\*\*\*\*\* DATA\_SOURCE \*\*\*\*\*\*\*\* ");

Console.WriteLine(" ¡¡ SUCCESFULL OBTAINING !!");

Console.WriteLine("");

Console.WriteLine("");

break;

case "2":

Console.Clear();

Console.WriteLine(" Press Any Key To Get All DATA SOURCES on DataBase");

Console.WriteLine("");

Console.ReadKey();

var getResponse0 = DataSourceClient.GetAllDataSources(new Google.Protobuf.WellKnownTypes.Empty());

if (getResponse0.Items is null)

{

Console.WriteLine(" Is NULL");

channel.Dispose();

return;

}

else

{

Console.WriteLine($" SuccesFull Obtaining [({getResponse0.Items.Count})] DATA SOURCES");

Console.WriteLine("");

}

break;

}

break;

case "2":

Console.WriteLine(" Select How Many Items Do You Requerid...\n");

Console.WriteLine("------------------------------------------\n\n");

Console.WriteLine(" 1- Get 1 Item By Id");

Console.WriteLine(" 2- Get All Items From DataBase\n\n\n");

switch (Console.ReadLine())

{

case "1":

Console.WriteLine("");

Console.WriteLine("");

Console.WriteLine($" Press Any Key To Get The COMMUNICATION CLIENTE With ID => [ {createResponse1.Id} ] ");

Console.WriteLine("");

Console.WriteLine(" Proccessing....");

Console.WriteLine("");

Console.ReadKey();

var getByIdResponse1 = CommunicationClientClient.GetCommunicationClient(new GetRequest() { Id = createResponse1.Id.ToString() });

if (getByIdResponse1 is null)

{

Console.WriteLine(" Cannot get COMMUNICATION CLIENT");

channel.Dispose();

return;

}

else

{

Console.WriteLine("");

}

Console.WriteLine(" \*\*\*\*\*\*\*\* COMMUNICATION\_CLIENT \*\*\*\*\*\*\*\* ");

Console.WriteLine(" ¡¡ SUCCESFULL OBTAINING !!");

Console.WriteLine("");

Console.WriteLine("");

break;

case "2":

Console.WriteLine(" Press Any Key To Get All COMMUNICATION CLIENTS on DataBase");

Console.WriteLine("");

Console.ReadKey();

var getResponse1 = CommunicationClientClient.GetAllCommunicationClients(new Google.Protobuf.WellKnownTypes.Empty());

if (getResponse1.Items is null)

{

Console.WriteLine(" Is NULL");

channel.Dispose();

return;

}

else

{

Console.WriteLine($" SuccesFull Obtaining [({getResponse1.Items.Count})] COMMUNICATION CLIENTS");

Console.WriteLine("");

}

break;

}

break;

case "3":

Console.WriteLine(" Select How Many Items Do You Requerid...\n");

Console.WriteLine("------------------------------------------\n\n");

Console.WriteLine(" 1- Get 1 Item By Id");

Console.WriteLine(" 2- Get All Items From DataBase\n\n\n");

switch (Console.ReadLine())

{

case "1":

Console.WriteLine("");

Console.WriteLine("");

Console.WriteLine($" Press Any Key To Get The OPC NODE With ID => [ {createResponse2.Id} ] ");

Console.WriteLine("");

Console.WriteLine(" Proccessing....");

Console.WriteLine("");

Console.ReadKey();

var getByIdResponse2 = OPCNodeClient.GetOPCNode(new GetRequest() { Id = createResponse2.Id.ToString() });

if (getByIdResponse2 is null)

{

Console.WriteLine(" Cannot get OPC NODE");

channel.Dispose();

return;

}

else

{

Console.WriteLine("");

}

Console.WriteLine(" \*\*\*\*\*\*\*\* OPC\_NODE \*\*\*\*\*\*\*\* ");

Console.WriteLine(" ¡¡ SUCCESFULL OBTAINING !!");

Console.WriteLine("");

Console.WriteLine("");

break;

case "2":

Console.WriteLine(" Press Any Key To Get All OPC\_NODES on DataBase");

Console.WriteLine("");

Console.ReadKey();

var getResponse2 = OPCNodeClient.GetAllOPCNodes(new Google.Protobuf.WellKnownTypes.Empty());

if (getResponse2.Items is null)

{

Console.WriteLine(" Is NULL");

channel.Dispose();

return;

}

else

{

Console.WriteLine($" SuccesFull Obtaining [({getResponse2.Items.Count})] OPC NODES");

Console.WriteLine("");

}

break;

}

break;

case "4":

Console.WriteLine(" Select How Many Items Do You Requerid...\n");

Console.WriteLine("------------------------------------------\n\n");

Console.WriteLine(" 1- Get 1 Item By Id");

Console.WriteLine(" 2- Get All Items From DataBase\n\n\n");

switch (Console.ReadLine())

{

case "1":

Console.WriteLine("");

Console.WriteLine("");

Console.WriteLine($" Press Any Key To Get The MODBUS NODE With ID => [ {createResponse3.Id} ] ");

Console.WriteLine("");

Console.WriteLine(" Proccessing....");

Console.WriteLine("");

Console.ReadKey();

var getByIdResponse3 = ModbusNodeClient.GetModbusNode(new GetRequest() { Id = createResponse3.Id.ToString() });

if (getByIdResponse3 is null)

{

Console.WriteLine(" Cannot get MODBUS NODE");

channel.Dispose();

return;

}

else

{

Console.WriteLine("");

}

Console.WriteLine(" \*\*\*\*\*\*\*\* MODBUS\_NODE \*\*\*\*\*\*\*\* ");

Console.WriteLine(" ¡¡ SUCCESFULL OBTAINING !!");

Console.WriteLine("");

Console.WriteLine("");

break;

case "2":

Console.WriteLine(" Press Any Key To Get All MODBUS\_NODES on DataBase");

Console.WriteLine("");

Console.ReadKey();

var getResponse3 = ModbusNodeClient.GetAllModbusNodes(new Google.Protobuf.WellKnownTypes.Empty());

if (getResponse3.Items is null)

{

Console.WriteLine(" Is NULL");

channel.Dispose();

return;

}

else

{

Console.WriteLine($" SuccesFull Obtaining [({getResponse3.Items.Count})] MODBUS NODES");

Console.WriteLine("");

}

break;

}

break;

default:

Console.WriteLine("Leaving");

break;

}

break;

case "3":

switch (DateTypeSelect)

{

case "1":

#region UPDATE DATA SOURCE

Console.Clear();

Console.WriteLine(" Press Any Key To Update The DATA SOURCE");

Console.WriteLine("");

Console.WriteLine(" Updating....");

Console.ReadKey();

createResponse0.InputsCounter = 5;

DataSourceClient.UpdateDataSource(createResponse0);

var updatedGetResponse0 = DataSourceClient.GetDataSource(new GetRequest() { Id = createResponse0.Id });

if (updatedGetResponse0 is not null &&

updatedGetResponse0.KindCase == NullableDataSourceDTO.KindOneofCase.DataSource &&

updatedGetResponse0.DataSource.InputsCounter == createResponse0.InputsCounter)

{

Console.WriteLine("");

Console.WriteLine("");

Console.WriteLine(" \*\*\*\*\*\*\*\* DATA\_SOURCE \*\*\*\*\*\*\*\* ");

Console.WriteLine(" ¡¡ SUCCESFULL UPDATE !!");

Console.WriteLine("");

Console.WriteLine("");

}

#endregion

break;

case "2":

#region UPDATE COMMUNICATION CLIENT

Console.WriteLine(" Press Any Key To Update The COMMUNICATION CLIENT");

Console.WriteLine("");

Console.WriteLine(" Updating....");

Console.ReadKey();

createResponse1.ConnectionPort = "100.23.55.55";

CommunicationClientClient.UpdateCommunicationClient(createResponse1);

var updatedGetResponse1 = CommunicationClientClient.GetCommunicationClient(new GetRequest() { Id = createResponse1.Id });

if (updatedGetResponse1 is not null &&

updatedGetResponse1.KindCase == NullableCommunicationClientDTO.KindOneofCase.CommunicationClient &&

updatedGetResponse1.CommunicationClient.ConnectionPort == createResponse1.ConnectionPort)

{

Console.WriteLine("");

Console.WriteLine("");

Console.WriteLine(" \*\*\*\*\*\*\*\* COMMUNICATION CLIENT \*\*\*\*\*\*\*\* ");

Console.WriteLine(" ¡¡ SUCCESFULL UPDATE !!");

Console.WriteLine("");

Console.WriteLine("");

}

#endregion

break;

case "3":

#region UPDATE OPC NODE

Console.WriteLine(" Press Any Key To Update The OPC NODE");

Console.WriteLine("");

Console.WriteLine(" Updating....");

Console.ReadKey();

createResponse2.AddressLabel = "OPC\_NODE\_1.0.1";

OPCNodeClient.UpdateOPCNode(createResponse2);

var updatedGetResponse2 = OPCNodeClient.GetOPCNode(new GetRequest() { Id = createResponse2.Id });

if (updatedGetResponse2 is not null &&

updatedGetResponse2.KindCase == NullableOPCNodeDTO.KindOneofCase.OpcNode &&

updatedGetResponse2.OpcNode.AddressLabel == createResponse2.AddressLabel)

{

Console.WriteLine("");

Console.WriteLine("");

Console.WriteLine(" \*\*\*\*\*\*\*\* OPC NODE \*\*\*\*\*\*\*\* ");

Console.WriteLine(" ¡¡ SUCCESFULL UPDATE !!");

Console.WriteLine("");

Console.WriteLine("");

}

#endregion

break;

case "4":

#region UPDATE MODBUS NODE

Console.WriteLine(" Press Any Key To Update The MODBUS NODE");

Console.WriteLine("");

Console.WriteLine(" Updating....");

Console.ReadKey();

createResponse3.RecordSource = 2;

ModbusNodeClient.UpdateModbusNode(createResponse3);

var updatedGetResponse3 = ModbusNodeClient.GetModbusNode(new GetRequest() { Id = createResponse3.Id });

if (updatedGetResponse3 is not null &&

updatedGetResponse3.KindCase == NullableModbusNodeDTO.KindOneofCase.ModbusNode &&

updatedGetResponse3.ModbusNode.RecordSource == createResponse3.RecordSource)

{

Console.WriteLine("");

Console.WriteLine("");

Console.WriteLine(" \*\*\*\*\*\*\*\* MODBUS NODE \*\*\*\*\*\*\*\* ");

Console.WriteLine(" ¡¡ SUCCESFULL UPDATE !!");

Console.WriteLine("");

Console.WriteLine("");

}

#endregion

break;

default:

Console.WriteLine(" Finishing");

cicle = false;

break;

}

break;

case "4":

switch (DateTypeSelect)

{

case "1":

#region DELETE DATA SOURCE

Console.Clear();

Console.WriteLine("");

Console.WriteLine(" Press Any Key To Delete The DATA SOURCE");

Console.WriteLine("");

Console.WriteLine(" Eliminating....");

Console.ReadKey();

Console.WriteLine("");

DataSourceClient.DeleteDataSource(new DeleteRequest() { Id = createResponse0.Id });

var deletedGetResponse0 = DataSourceClient.GetDataSource(new GetRequest() { Id = createResponse0.Id });

if (deletedGetResponse0 is null ||

deletedGetResponse0.KindCase != NullableDataSourceDTO.KindOneofCase.DataSource)

{

Console.WriteLine(" \*\*\*\*\*\*\*\* DATA\_SOURCE \*\*\*\*\*\*\*\* ");

Console.WriteLine(" ¡¡ SUCCESFULL DELETE !!");

Console.WriteLine("");

Console.WriteLine("");

}

#endregion

break;

case "2":

#region DELETE COMMUNICATION CLIENT

Console.WriteLine("");

Console.WriteLine(" Press Any Key To Delete The COMMUNICATION CLIENT");

Console.WriteLine("");

Console.WriteLine(" Eliminating....");

Console.ReadKey();

Console.WriteLine("");

CommunicationClientClient.DeleteCommunicationClient(new DeleteRequest() { Id = createResponse1.Id });

var deletedGetResponse1 = CommunicationClientClient.GetCommunicationClient(new GetRequest() { Id = createResponse1.Id });

if (deletedGetResponse1 is null ||

deletedGetResponse1.KindCase != NullableCommunicationClientDTO.KindOneofCase.CommunicationClient)

{

Console.WriteLine(" \*\*\*\*\*\*\*\* COMMUNICATION CLIENT \*\*\*\*\*\*\*\* ");

Console.WriteLine(" ¡¡ SUCCESFULL DELETE !!");

Console.WriteLine("");

Console.WriteLine("");

}

#endregion

break;

case "3":

#region DELETE OPC NODE

Console.WriteLine("");

Console.WriteLine(" Press Any Key To Delete The OPC NODE");

Console.WriteLine("");

Console.WriteLine(" Eliminating....");

Console.ReadKey();

Console.WriteLine("");

OPCNodeClient.DeleteOPCNode(new DeleteRequest() { Id = createResponse2.Id });

var deletedGetResponse2 = OPCNodeClient.GetOPCNode(new GetRequest() { Id = createResponse2.Id });

if (deletedGetResponse2 is null ||

deletedGetResponse2.KindCase != NullableOPCNodeDTO.KindOneofCase.OpcNode)

{

Console.WriteLine(" \*\*\*\*\*\*\*\* OPC NODE \*\*\*\*\*\*\*\* ");

Console.WriteLine(" ¡¡ SUCCESFULL DELETE !!");

Console.WriteLine("");

Console.WriteLine("");

}

#endregion

break;

case "4":

#region DELETE MODBUS NODE

Console.WriteLine("");

Console.WriteLine(" Press Any Key To Delete The MODBUS NODE");

Console.WriteLine("");

Console.WriteLine(" Eliminating....");

Console.ReadKey();

Console.WriteLine("");

ModbusNodeClient.DeleteModbusNode(new DeleteRequest() { Id = createResponse3.Id });

var deletedGetResponse3 = ModbusNodeClient.GetModbusNode(new GetRequest() { Id = createResponse3.Id });

if (deletedGetResponse3 is null ||

deletedGetResponse3.KindCase != NullableModbusNodeDTO.KindOneofCase.ModbusNode)

{

Console.WriteLine(" \*\*\*\*\*\*\*\* MODBUS NODE \*\*\*\*\*\*\*\* ");

Console.WriteLine(" ¡¡ SUCCESFULL DELETE !!");

Console.WriteLine("");

Console.WriteLine("");

Console.WriteLine("");

Console.WriteLine("");

Console.WriteLine("");

Console.WriteLine("");

Console.WriteLine("");

Console.WriteLine("");

}

#endregion

break;

}

break;

case "5":

Console.WriteLine(" Finishing");

cicle = false;

break;

}

}

}

else

{

Console.WriteLine("");

Console.WriteLine("");

Console.WriteLine(" Invalid Password Try Again.....");

}

} while (Try <= 3);

#endregion

#endregion

#region \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* CRUDs \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

#region CREATE

//Console.WriteLine("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* CREATE \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

//Console.WriteLine("");

//Console.WriteLine(" Creado con Exito");

//// Creando Fuentes de datos

//DataSource dataSource1 = new DataSource

// (Guid.NewGuid(), "code1", 3, 2);

//DataSource dataSource2 = new DataSource

// (Guid.NewGuid(), "code2", 1, 1);

//// Creando Clientes de Comunicacion

//CommunicationClient communicationClient1 = new CommunicationClient

// (Guid.NewGuid(), "68.96.88.4", dataSource1);

//CommunicationClient communicationClient2 = new CommunicationClient

// (Guid.NewGuid(), "88.100.48.2", dataSource2);

//// Creando Nodos Modbus (Hardwares)

//ModbusNode modbusNode1 = new ModbusNode

// (Guid.NewGuid(), "ModBus\_1", 1, communicationClient1);

//ModbusNode modbusNode2 = new ModbusNode

// (Guid.NewGuid(), "ModBus\_2", 3, communicationClient2);

//// Creando Nodos OPC (Softwares)

//OPCNode oPCNode1 = new OPCNode

// (Guid.NewGuid(), "68.96.88.4", communicationClient1);

//OPCNode oPCNode2 = new OPCNode

// (Guid.NewGuid(), "88.100.48.2", communicationClient2);

//// SALVA en BASE DATOS

////Añadiendo Fuentes de Datos a BD

//applicationContext.Set<DataSource>().Add(dataSource1);

//applicationContext.Set<DataSource>().Add(dataSource2);

//// Guardar Modificaciones Realizadas

//applicationContext.SaveChanges();

////Añadiendo Clientes de Comunicacion a BD

//applicationContext.Set<CommunicationClient>().Add(communicationClient1);

//applicationContext.Set<CommunicationClient>().Add(communicationClient2);

//// Guardar Modificaciones Realizadas

//applicationContext.SaveChanges();

////Añadiendo Nodos Modbus a BD

//applicationContext.Set<ModbusNode>().Add(modbusNode1);

//applicationContext.Set<ModbusNode>().Add(modbusNode2);

//// Guardar Modificaciones Realizadas

//applicationContext.SaveChanges();

////Añadiendo Nodos OPC a BD

//applicationContext.Set<OPCNode>().Add(oPCNode1);

//applicationContext.Set<OPCNode>().Add(oPCNode2);

//// Guardar Modificaciones Realizadas

//applicationContext.SaveChanges();

#endregion

#region READ

//Console.WriteLine("");

//Console.WriteLine("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* GET ID \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

//Console.WriteLine("");

//Console.WriteLine("");

//// Obteniendo ID desde Base Datos

//DataSource? dataSourceOfClient1 = applicationContext

// .Set<DataSource>()

// .FirstOrDefault(d => d.Id == communicationClient1.DataSourceId);

//// Obteniendo ID desde Base Datos

//DataSource? dataSourceOfClient2 = applicationContext

// .Set<DataSource>()

// .FirstOrDefault(d => d.Id == communicationClient2.DataSourceId);

//// Obteniendo ID desde Base Datos

//CommunicationClient? communicationClientOfModbusNodes1 = applicationContext

// .Set<CommunicationClient>()

// .FirstOrDefault(c => c.Id == modbusNode1.CommunicationClientId);

//// Obteniendo ID desde Base Datos

//CommunicationClient? communicationClientOfModbusNodes2 = applicationContext

// .Set<CommunicationClient>()

// .FirstOrDefault(c => c.Id == modbusNode2.CommunicationClientId);

//// Obteniendo ID desde Base Datos

//CommunicationClient? communicationClientOfOPCNodes1 = applicationContext

// .Set<CommunicationClient>()

// .FirstOrDefault(c => c.Id == oPCNode1.CommunicationClientId);

//// Obteniendo ID desde Base Datos

//CommunicationClient? communicationClientOfOPCNodes2 = applicationContext

// .Set<CommunicationClient>()

// .FirstOrDefault(c => c.Id == oPCNode2.CommunicationClientId);

//Console.WriteLine("Identificadores obtenidos con EXITO");

//Console.WriteLine("");

//Console.WriteLine("");

//Console.WriteLine("");

#endregion

#region UPDATE

//Console.WriteLine("");

//Console.WriteLine("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* UPDATE \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

//Console.WriteLine("");

//// Modificando Aspectos de Fuente de Datos en la Base de Datos

//var dataSources = applicationContext.Set<DataSource>().ToList();

//foreach (var loadedDataSource in dataSources)

//{

// loadedDataSource.DataSourceType = DataSourceType.HMI;

// applicationContext.Set<DataSource>().Update(loadedDataSource);

//}

//applicationContext.SaveChanges();

//Console.WriteLine("Nuevo tipo de Fuente de Datos");

//// Modificando Aspectos del Cliente de Comunicacion en la Base de Datos

//var communicationClients = applicationContext.Set<CommunicationClient>().ToList();

//foreach (var loadedCommunicationClient in communicationClients)

//{

// loadedCommunicationClient.ConnectionPort = "8";

// applicationContext.Set<CommunicationClient>().Update(loadedCommunicationClient);

//}

//applicationContext.SaveChanges();

//Console.WriteLine("Nuevo Puerto de Conexion del Cliente de Comunicacion");

//// Modificando Aspectos del Nodo MODBUS en la Base de Datos

//var modbusNodes = applicationContext.Set<ModbusNode>().ToList();

//foreach (var loadedModbusNode in modbusNodes)

//{

// loadedModbusNode.RecordToRead = 2;

// loadedModbusNode.RecordSource = 4;

// applicationContext.Set<ModbusNode>().Update(loadedModbusNode);

//}

//applicationContext.SaveChanges();

//Console.WriteLine("Nuevo Registro para obtener info por Nodo Modbus ");

//// Modificando Aspectos del Nodo MODBUS en la Base de Datos

//var opcNodes = applicationContext.Set<OPCNode>().ToList();

//foreach (var loadedOPCNode in opcNodes)

//{

// loadedOPCNode.AddressLabel = "100.90.80.5";

// applicationContext.Set<OPCNode>().Update(loadedOPCNode);

//}

//applicationContext.SaveChanges();

//Console.WriteLine("Nueva direccion a consumir por Nodo OPC ");

#endregion

#region DELETE

//Console.WriteLine("");

//Console.WriteLine("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* DELETE \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

//Console.WriteLine("");

//// Eliminando de Base de Datos

//// Eliminando un Nodo OPC del Soporte de Datos

//applicationContext.CommunicationNodes.Remove(oPCNode1);

//applicationContext.SaveChanges();

//OPCNode? deletedOPCNode = applicationContext.Set<OPCNode>()

// .FirstOrDefault(o => o.Id == oPCNode1.Id);

//if (deletedOPCNode is null)

// Console.WriteLine("Nodo OPC eliminado con Exito");

//// Eliminando un Nodo Modbus del Soporte de Datos

//applicationContext.CommunicationNodes.Remove(modbusNode1);

//applicationContext.SaveChanges();

//ModbusNode? deletedModbusNode = applicationContext.Set<ModbusNode>()

// .FirstOrDefault(m => m.Id == modbusNode1.Id);

//if (deletedModbusNode is null)

// Console.WriteLine("Nodo MODBUS eliminado con Exito");

//// Eliminando un Cliente de Comunicacion del Soporte de Datos

//applicationContext.CommunicationClients.Remove(communicationClient1);

//applicationContext.SaveChanges();

//CommunicationClient? deletedCommunicationClient = applicationContext.Set<CommunicationClient>()

// .FirstOrDefault(c => c.Id == communicationClient1.Id);

//if (deletedCommunicationClient is null)

// Console.WriteLine("Cliente de Comunicacion eliminado con Exito");

//// Eliminando una Fuente de Datos del Soporte de Datos

//applicationContext.DataSources.Remove(dataSource1);

//applicationContext.SaveChanges();

//DataSource? deletedDataSource = applicationContext.Set<DataSource>()

// .FirstOrDefault(d => d.Id == dataSource1.Id);

//if (deletedDataSource is null)

// Console.WriteLine("Fuente de Datos eliminada con Exito");

//Console.WriteLine("");

//Console.WriteLine("");

//Console.WriteLine(""); //Console.WriteLine(""); //Console.WriteLine("");

//Console.WriteLine(" \_\_ FIN \_\_");

//Console.WriteLine(""); //Console.WriteLine(""); //Console.WriteLine(""); //Console.WriteLine("");

//Console.WriteLine("");

//Console.WriteLine("");

//Console.WriteLine("");

//Console.WriteLine("");

#endregion

#endregion

}

}

}