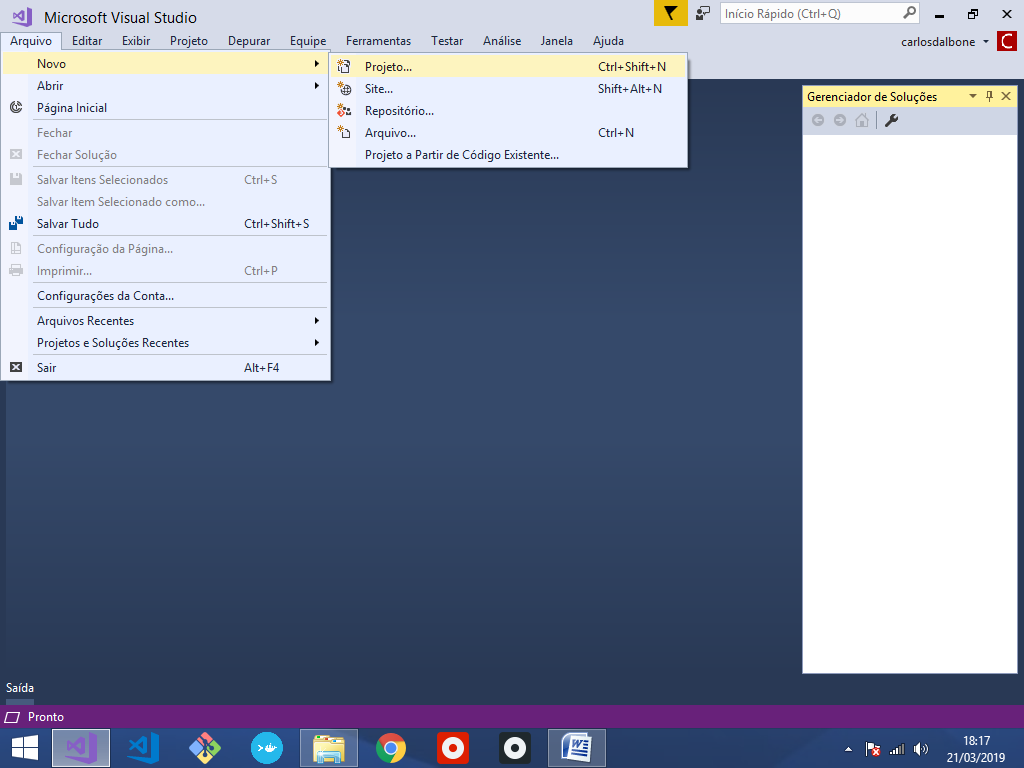
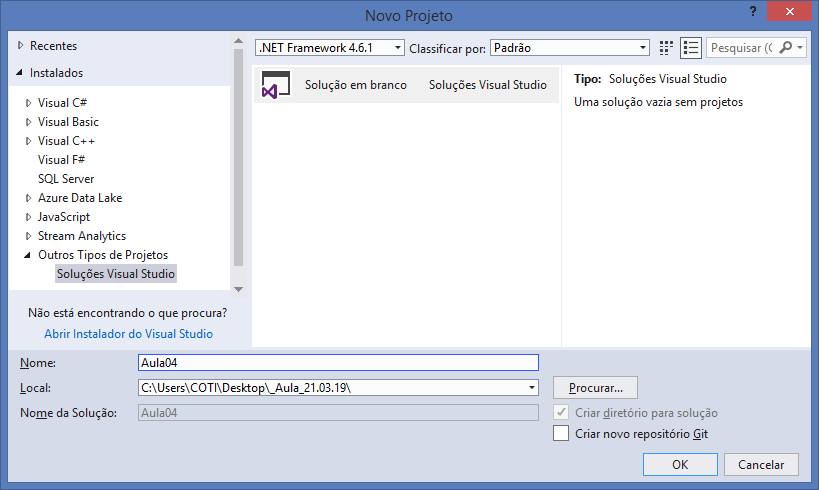
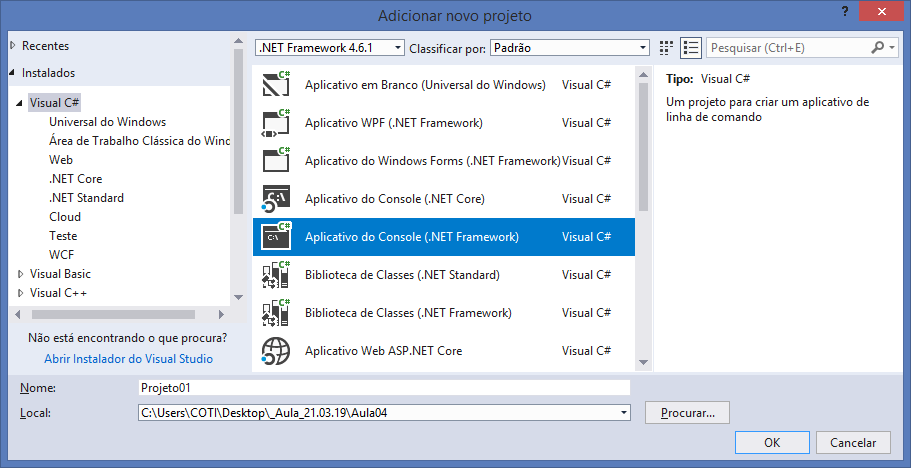
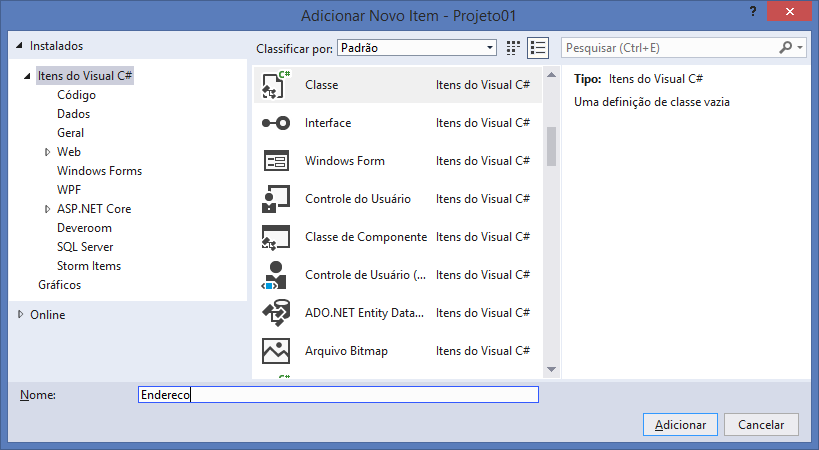
**Criando uma nova solution em branco:**









using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Projeto01.Entities

{

public class Endereco

{

//prop + 2x[tab]

public int IdEndereco { get; set; }

public string Logradouro { get; set; }

public string Bairro { get; set; }

public string Cidade { get; set; }

public string Estado { get; set; }

public string Cep { get; set; }

//ctor + 2x[tab]

public Endereco()

{

//construtor default (vazio)

}

//sobrecarga de método construtor (overloading)

public Endereco(int idEndereco, string logradouro,

string bairro, string cidade, string estado, string cep)

{

IdEndereco = idEndereco;

Logradouro = logradouro;

Bairro = bairro;

Cidade = cidade;

Estado = estado;

Cep = cep;

}

//sobrescrita do método ToString (override)

public override string ToString()

{

return $"Id: {IdEndereco}, Logradouro: {Logradouro},

Bairro: {Bairro}, "

+ $"Cidade: {Cidade}, Estado: {Estado}, Cep: {Cep}";

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Projeto01.Entities

{

public class Contato

{

public string Telefone { get; set; }

public string Email { get; set; }

public Contato()

{

}

public Contato(string telefone, string email)

{

Telefone = telefone;

Email = email;

}

public override string ToString()

{

return $"Telefone: {Telefone}, Email: {Email}";

}

}

}

**Criando um ENUM para relacionar com a classe Endereço:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Projeto01.Entities.Types

{

public enum TipoEndereco

{

Residencial = 1,

Comercial = 2

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using Projeto01.Entities.Types; //importando

namespace Projeto01.Entities

{

public class Endereco

{

//prop + 2x[tab]

public int IdEndereco { get; set; }

public string Logradouro { get; set; }

public string Bairro { get; set; }

public string Cidade { get; set; }

public string Estado { get; set; }

public string Cep { get; set; }

public TipoEndereco Tipo { get; set; }

//ctor + 2x[tab]

public Endereco()

{

//construtor default (vazio)

}

//sobrecarga de método construtor (overloading)

public Endereco(int idEndereco, string logradouro, string bairro,

string cidade, string estado, string cep, TipoEndereco tipo)

{

IdEndereco = idEndereco;

Logradouro = logradouro;

Bairro = bairro;

Cidade = cidade;

Estado = estado;

Cep = cep;

Tipo = tipo;

}

//sobrescrita do método ToString (override)

public override string ToString()

{

return $"Id: {IdEndereco}, Logradouro: {Logradouro},

Bairro: {Bairro}, "

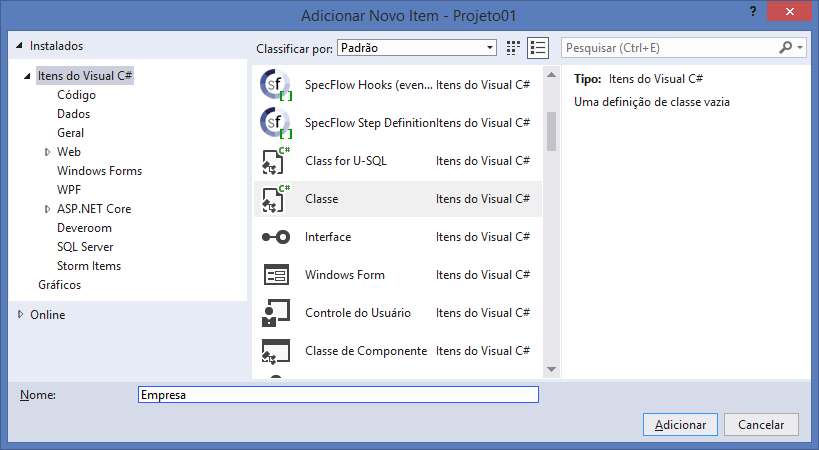
+ $"Cidade: {Cidade}, Estado: {Estado}, Cep: {Cep},

Tipo: {Tipo}";

}

}

}



using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Projeto01.Entities

{

public class Empresa

{

public int IdEmpresa { get; set; }

public string RazaoSocial { get; set; }

public string Cnpj { get; set; }

public Empresa()

{

}

public Empresa(int idEmpresa, string razaoSocial, string cnpj)

{

IdEmpresa = idEmpresa;

RazaoSocial = razaoSocial;

Cnpj = cnpj;

}

public override string ToString()

{

return $"Id: {IdEmpresa}, Razão Social: {RazaoSocial}, CNPJ: {Cnpj}";

}

}

}

**Definindo os relacionamentos:**

* Empresa TEM 1 Contato
* Empresa TEM MUITOS Enderecos

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Projeto01.Entities

{

public class Empresa

{

public int IdEmpresa { get; set; }

public string RazaoSocial { get; set; }

public string Cnpj { get; set; }

//Associação TER-1 (EMPRESA TEM 1 CONTATO)

**public Contato Contato { get; set; }**

//Associação TER-N (EMPRESA TEM MUITOS ENDERECOS)

**public List<Endereco> Enderecos { get; set; }**

public Empresa()

{

}

public Empresa(int idEmpresa, string razaoSocial, string cnpj)

{

IdEmpresa = idEmpresa;

RazaoSocial = razaoSocial;

Cnpj = cnpj;

}

public override string ToString()

{

return $"Id: {IdEmpresa}, Razão Social: {RazaoSocial}, CNPJ: {Cnpj}";

}

}

}

**Voltando no método Main():**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using Projeto01.Entities; //importando

using Projeto01.Entities.Types; //importando

namespace Projeto01

{

class Program

{

static void Main(string[] args)

{

Empresa empresa = new Empresa();

empresa.Contato = new Contato();

empresa.Enderecos = new List<Endereco>();

empresa.IdEmpresa = 1;

empresa.RazaoSocial = "COTI Informática";

empresa.Cnpj = "123456-7890";

empresa.Contato.Telefone = "(21)2262-9043";

empresa.Contato.Email = "contato@cotiinformatica.com.br";

Endereco endereco1 = new Endereco();

endereco1.IdEndereco = 1;

endereco1.Logradouro = "Av Rio Branco, 185";

endereco1.Bairro = "Centro";

endereco1.Cidade = "Rio de Janeiro";

endereco1.Estado = "RJ";

endereco1.Cep = "25000-000";

endereco1.Tipo = TipoEndereco.Comercial;

Endereco endereco2 = new Endereco();

endereco2.IdEndereco = 2;

endereco2.Logradouro = "Av Presidente Vargas, 100";

endereco2.Bairro = "Centro";

endereco2.Cidade = "Rio de Janeiro";

endereco2.Estado = "RJ";

endereco2.Cep = "24000-000";

endereco2.Tipo = TipoEndereco.Comercial;

//incluindo os endereços na Empresa..

empresa.Enderecos.Add(endereco1);

empresa.Enderecos.Add(endereco2);

Console.ReadKey(); //pausar..

}

}

}

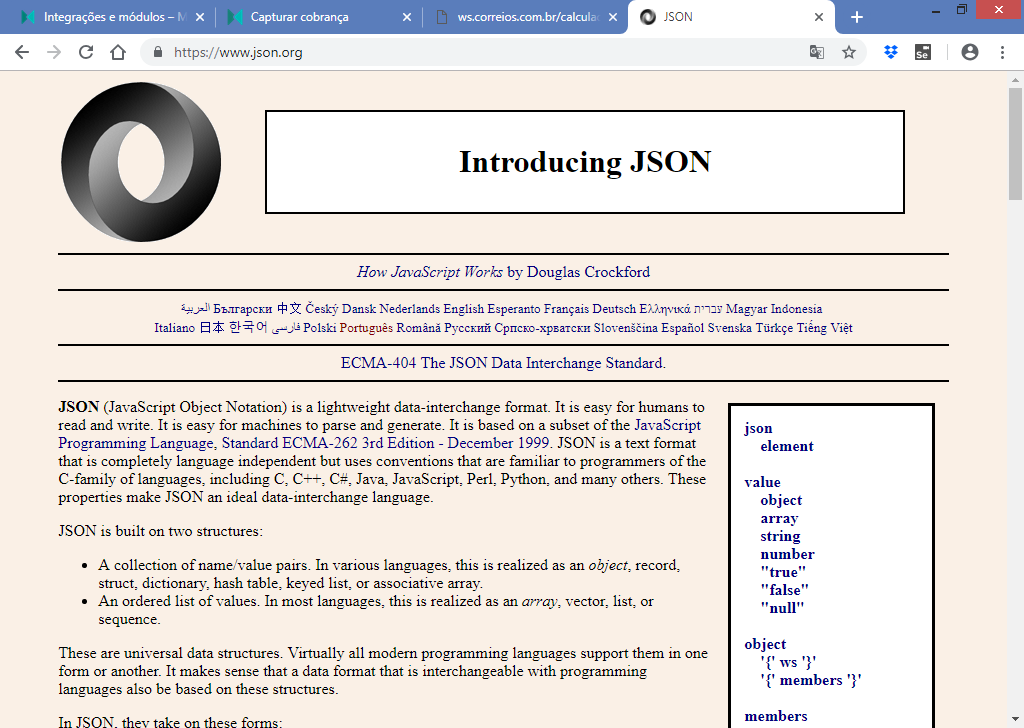
JSON JavaScript Object Notation

Notação de javascript orientado a objetos.

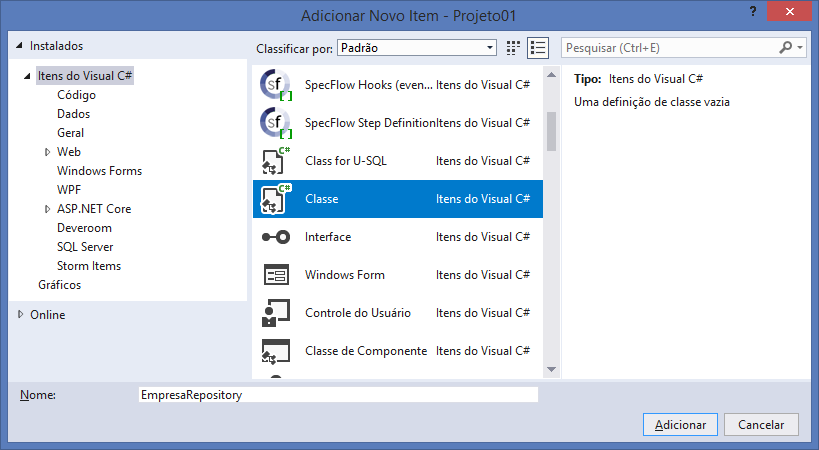
Formato de dados que tem como características:

* Mais simples e leve do que o XML
* Compatibilidade direta com JavaScript
* Utilizado para sistemas que fazem integração   
  utilizando API ao invés de WebServices (XML)

https://www.json.org/







using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using Projeto01.Entities; //importando

using System.IO; //importando

namespace Projeto01.Repositories

{

public class EmpresaRepository

{

public void ExportarJson(Empresa empresa)

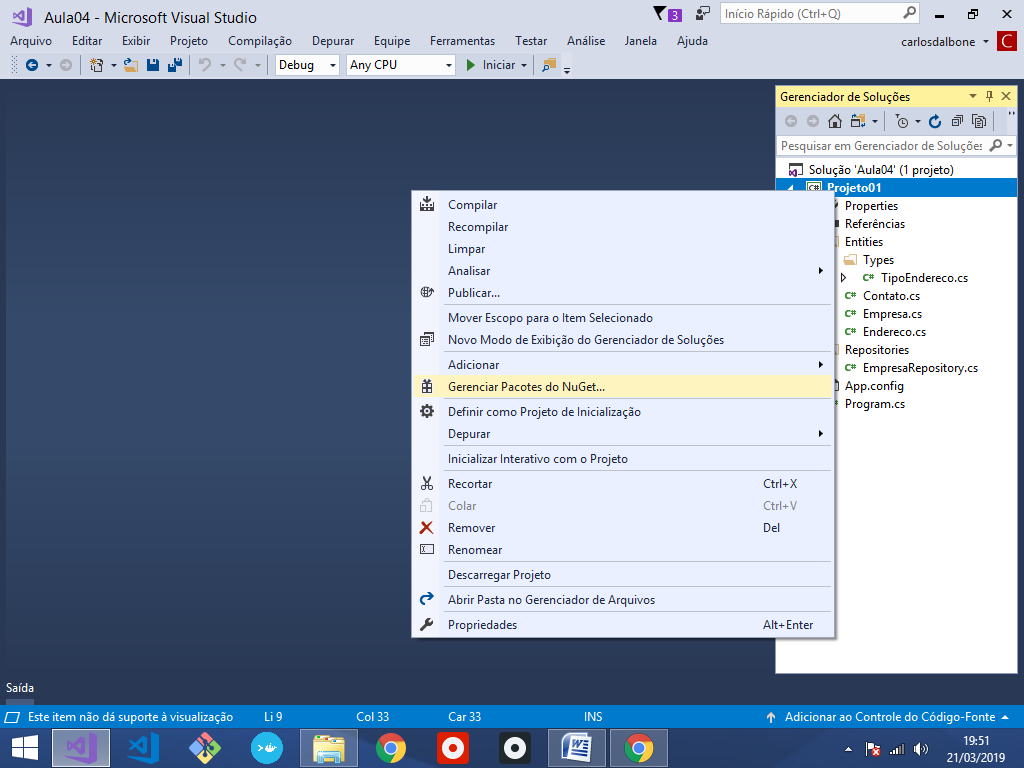
{

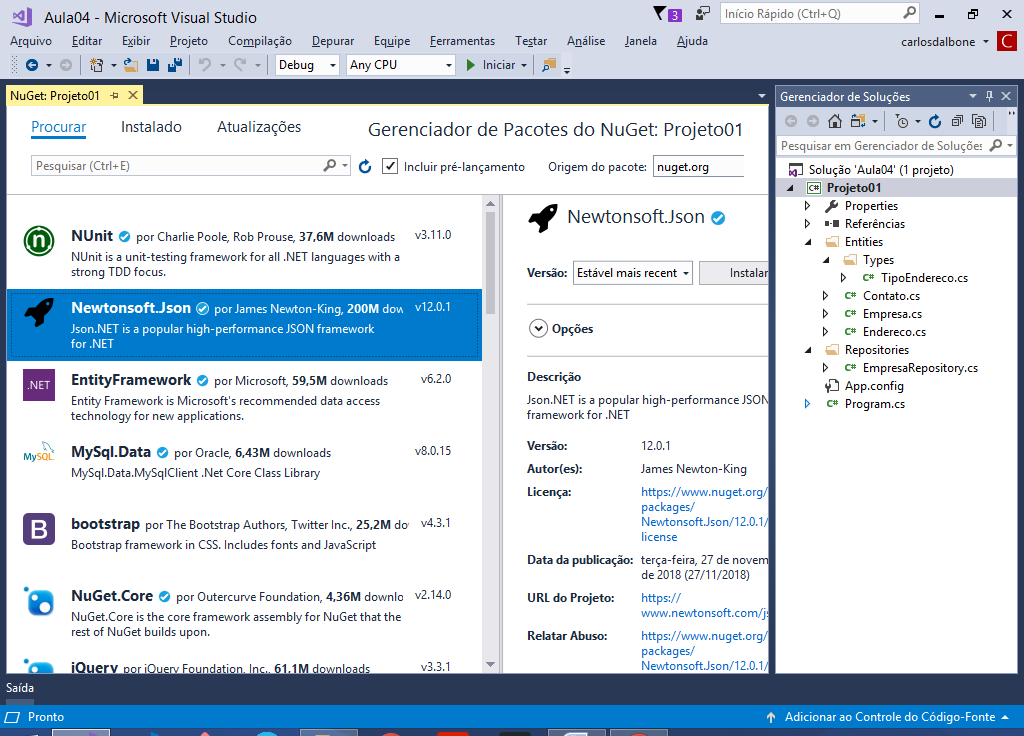
}

}

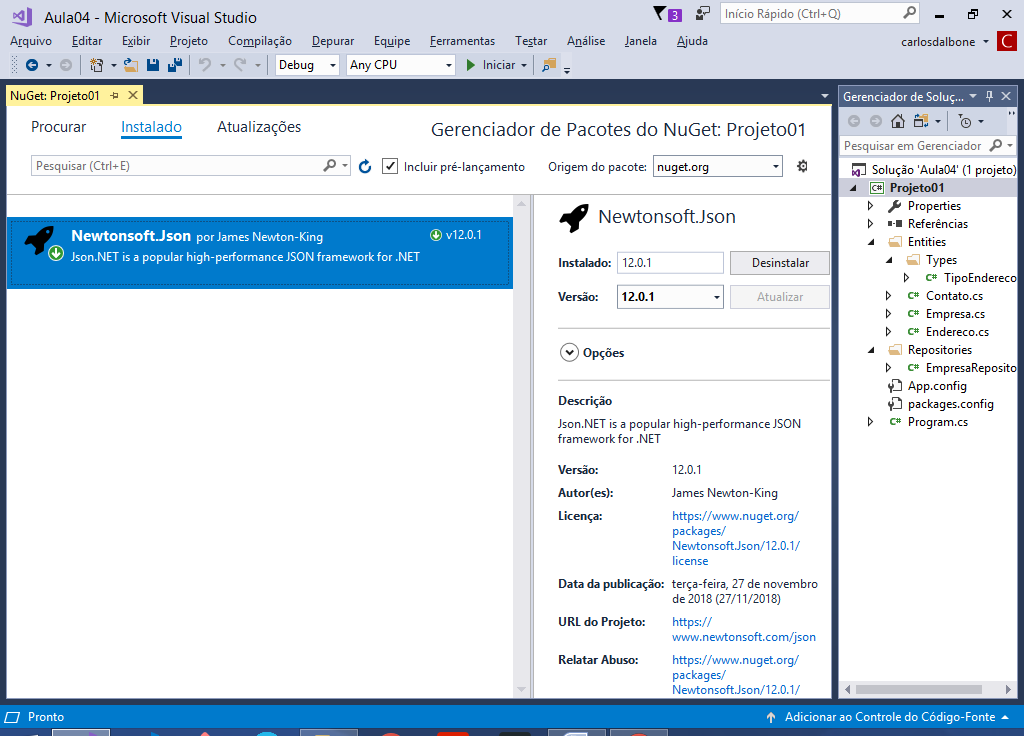
}

Para que possamos instalar a biblioteca necessária para produzir conteúdo em formato JSON, iremos utilizar o recurso do **NuGet** do VisualStudio





Framework instalado:



using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using Projeto01.Entities; //importando

using System.IO; //importando

using Newtonsoft.Json; //importando

namespace Projeto01.Repositories

{

public class EmpresaRepository

{

public void ExportarJson(Empresa empresa)

{

//abrindo um arquivo em modo de escrita

using (StreamWriter streamWriter

= new StreamWriter("c:\\temp\\empresa.json"))

{

//serializar o objeto 'empresa' para formato JSON

string dados = JsonConvert.SerializeObject

(empresa, Formatting.Indented);

//escrever no arquivo..

streamWriter.WriteLine(dados);

}

}

}

}

**Testando:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using Projeto01.Entities; //importando

using Projeto01.Entities.Types; //importando

using Projeto01.Repositories; //importando

namespace Projeto01

{

class Program

{

static void Main(string[] args)

{

Empresa empresa = new Empresa();

empresa.Contato = new Contato();

empresa.Enderecos = new List<Endereco>();

empresa.IdEmpresa = 1;

empresa.RazaoSocial = "COTI Informática";

empresa.Cnpj = "123456-7890";

empresa.Contato.Telefone = "(21)2262-9043";

empresa.Contato.Email = "contato@cotiinformatica.com.br";

Endereco endereco1 = new Endereco();

endereco1.IdEndereco = 1;

endereco1.Logradouro = "Av Rio Branco, 185";

endereco1.Bairro = "Centro";

endereco1.Cidade = "Rio de Janeiro";

endereco1.Estado = "RJ";

endereco1.Cep = "25000-000";

endereco1.Tipo = TipoEndereco.Comercial;

Endereco endereco2 = new Endereco();

endereco2.IdEndereco = 2;

endereco2.Logradouro = "Av Presidente Vargas, 100";

endereco2.Bairro = "Centro";

endereco2.Cidade = "Rio de Janeiro";

endereco2.Estado = "RJ";

endereco2.Cep = "24000-000";

endereco2.Tipo = TipoEndereco.Comercial;

//incluindo os endereços na Empresa..

empresa.Enderecos.Add(endereco1);

empresa.Enderecos.Add(endereco2);

**try**

**{**

**EmpresaRepository empresaRepository = new EmpresaRepository();**

**empresaRepository.ExportarJson(empresa);**

**Console.WriteLine("\nARQUIVO JSON GRAVADO COM SUCESSO.\n");**

**}**

**catch(Exception e)**

**{**

**Console.WriteLine("Erro: " + e.Message);**

**}**

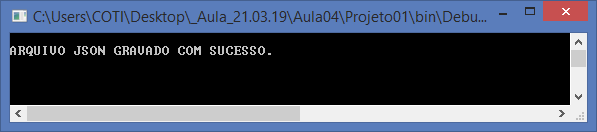
Console.ReadKey(); //pausar..

}

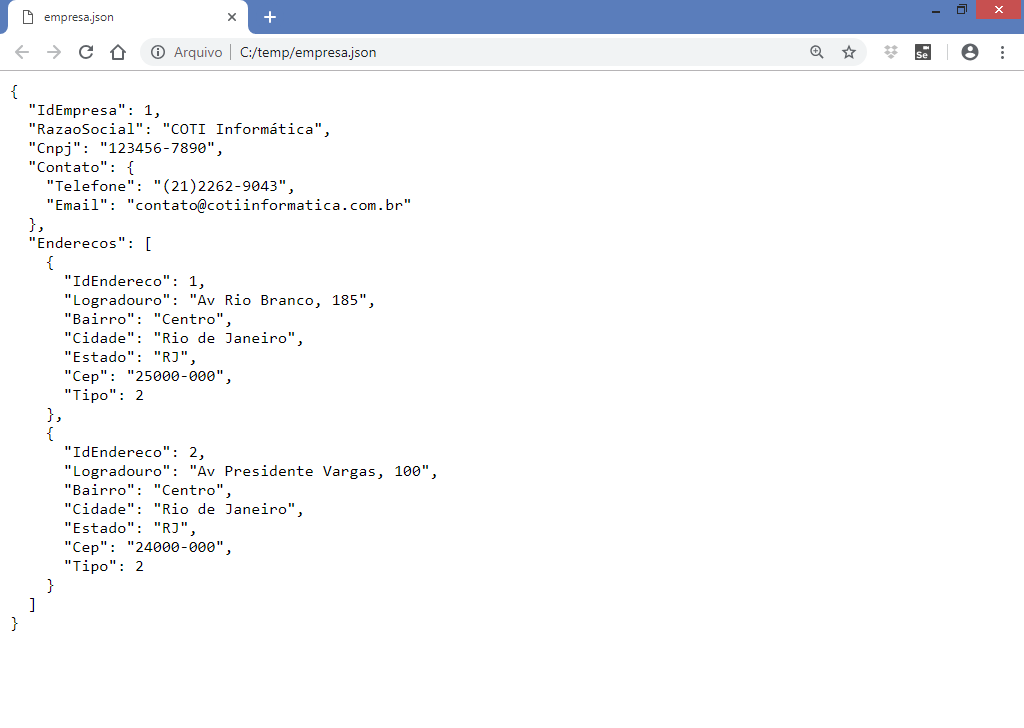
}

}

Executando:



**Arquivo gerado:**



---------------------------

{

"IdEmpresa": 1,

"RazaoSocial": "COTI Informática",

"Cnpj": "123456-7890",

"Contato": {

"Telefone": "(21)2262-9043",

"Email": "contato@cotiinformatica.com.br"

},

"Enderecos": [

{

"IdEndereco": 1,

"Logradouro": "Av Rio Branco, 185",

"Bairro": "Centro",

"Cidade": "Rio de Janeiro",

"Estado": "RJ",

"Cep": "25000-000",

"Tipo": 2

},

{

"IdEndereco": 2,

"Logradouro": "Av Presidente Vargas, 100",

"Bairro": "Centro",

"Cidade": "Rio de Janeiro",

"Estado": "RJ",

"Cep": "24000-000",

"Tipo": 2

}

]

}

Definindo o formato de **encoding** do arquivo:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using Projeto01.Entities; //importando

using System.IO; //importando

using Newtonsoft.Json; //importando

namespace Projeto01.Repositories

{

public class EmpresaRepository

{

public void ExportarJson(Empresa empresa)

{

//definindo o encoding do arquivo

**Encoding encoding = Encoding.UTF8;**

//abrindo um arquivo em modo de escrita

using (StreamWriter streamWriter

= new StreamWriter("c:\\temp\\empresa.json", false, **encoding**))

{

//serializar o objeto 'empresa' para formato JSON

string dados = JsonConvert.SerializeObject

(empresa, Formatting.Indented);

//escrever no arquivo..

streamWriter.WriteLine(dados);

}

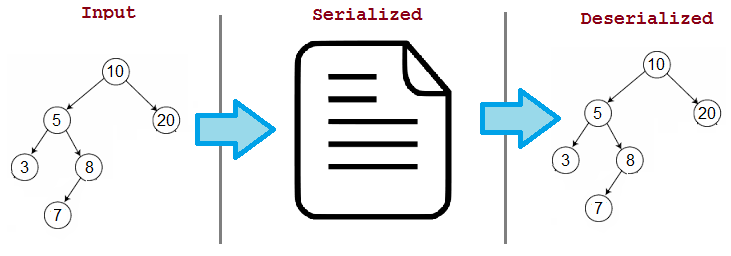
}

}

}

Criando um método para ler o conteúdo do arquivo

JSON e deserializar para o objeto Empresa:



using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using Projeto01.Entities; //importando

using System.IO; //importando

using Newtonsoft.Json; //importando

namespace Projeto01.Repositories

{

public class EmpresaRepository

{

public void ExportarJson(Empresa empresa)

{

//definindo o encoding do arquivo

Encoding encoding = Encoding.UTF8;

//abrindo um arquivo em modo de escrita

using (StreamWriter streamWriter

= new StreamWriter("c:\\temp\\empresa.json", false, encoding))

{

//serializar o objeto 'empresa' para formato JSON

string dados = JsonConvert.SerializeObject

(empresa, Formatting.Indented);

//escrever no arquivo..

streamWriter.WriteLine(dados);

}

}

**public Empresa ImportarJson()**

**{**

**//abrindo um arquivo em modo de leitura**

**using (StreamReader streamReader**

**= new StreamReader("c:\\temp\\empresa.json"))**

**{**

**//ler todo o texto contido no arquivo**

**string dados = streamReader.ReadToEnd();**

**//deserializar o conteudo JSON para objeto 'empresa'**

**Empresa empresa = JsonConvert.DeserializeObject<Empresa>(dados);**

**//retornar..**

**return empresa;**

**}**

**}**

}

}

**Testando:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using Projeto01.Entities; //importando

using Projeto01.Entities.Types; //importando

using Projeto01.Repositories; //importando

namespace Projeto01

{

class Program

{

static void Main(string[] args)

{

Empresa empresa = new Empresa();

empresa.Contato = new Contato();

empresa.Enderecos = new List<Endereco>();

empresa.IdEmpresa = 1;

empresa.RazaoSocial = "COTI Informática";

empresa.Cnpj = "123456-7890";

empresa.Contato.Telefone = "(21)2262-9043";

empresa.Contato.Email = "contato@cotiinformatica.com.br";

Endereco endereco1 = new Endereco();

endereco1.IdEndereco = 1;

endereco1.Logradouro = "Av Rio Branco, 185";

endereco1.Bairro = "Centro";

endereco1.Cidade = "Rio de Janeiro";

endereco1.Estado = "RJ";

endereco1.Cep = "25000-000";

endereco1.Tipo = TipoEndereco.Comercial;

Endereco endereco2 = new Endereco();

endereco2.IdEndereco = 2;

endereco2.Logradouro = "Av Presidente Vargas, 100";

endereco2.Bairro = "Centro";

endereco2.Cidade = "Rio de Janeiro";

endereco2.Estado = "RJ";

endereco2.Cep = "24000-000";

endereco2.Tipo = TipoEndereco.Comercial;

//incluindo os endereços na Empresa..

empresa.Enderecos.Add(endereco1);

empresa.Enderecos.Add(endereco2);

try

{

EmpresaRepository empresaRepository = new EmpresaRepository();

empresaRepository.ExportarJson(empresa);

Console.WriteLine("\nARQUIVO JSON GRAVADO COM SUCESSO.\n");

//ler o conteudo do arquivo..

Empresa resultado = empresaRepository.ImportarJson();

//imprimindo

Console.WriteLine("Empresa....: "

+ resultado.ToString());

Console.WriteLine("Contato....: "

+ resultado.Contato.ToString());

//varrer e imprimir os endereços

foreach(Endereco endereco in resultado.Enderecos)

{

Console.WriteLine("Endereço...: " + endereco.ToString());

}

}

catch(Exception e)

{

Console.WriteLine("Erro: " + e.Message);

}

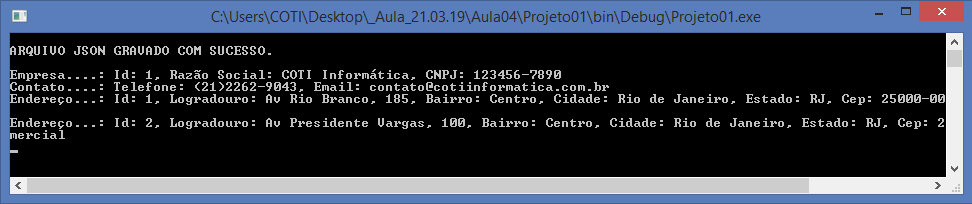
Console.ReadKey(); //pausar..

}

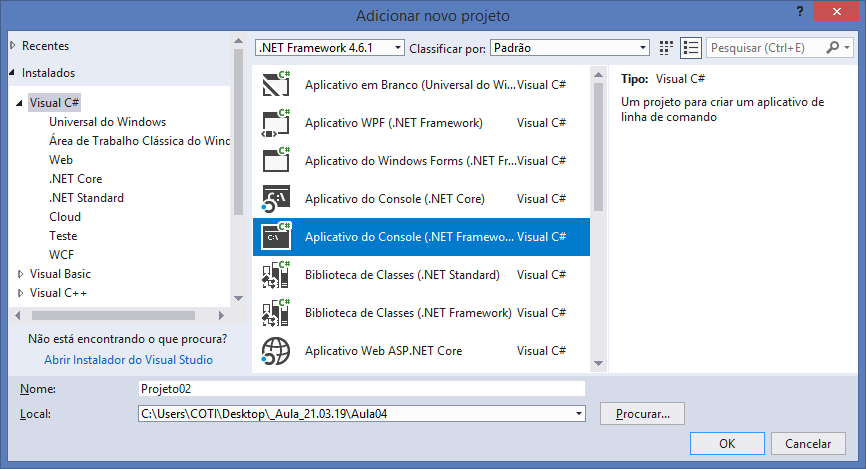
}

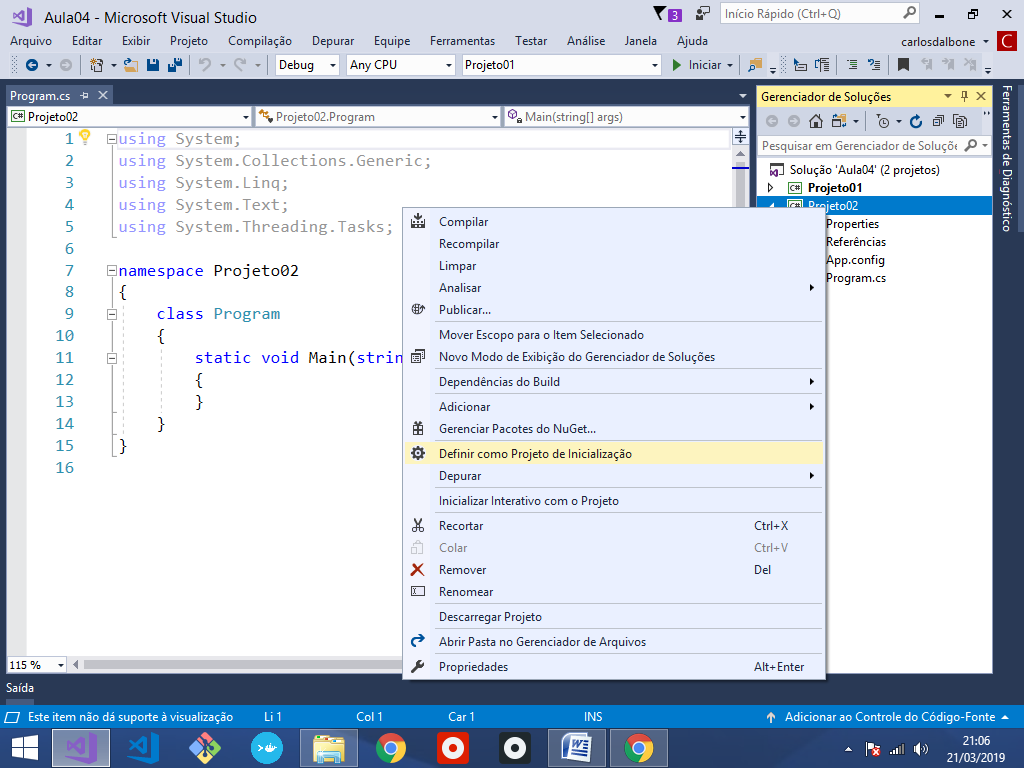
}

------------------------------



**Novo projeto:**





using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Projeto02.Entities

{

public class Cliente

{

public int IdCliente { get; set; }

public string Nome { get; set; }

public string Email { get; set; }

public Cliente()

{

}

public Cliente(int idCliente, string nome, string email)

{

IdCliente = idCliente;

Nome = nome;

Email = email;

}

public override string ToString()

{

return $"Id: {IdCliente}, Nome: {Nome}, Email: {Email}";

}

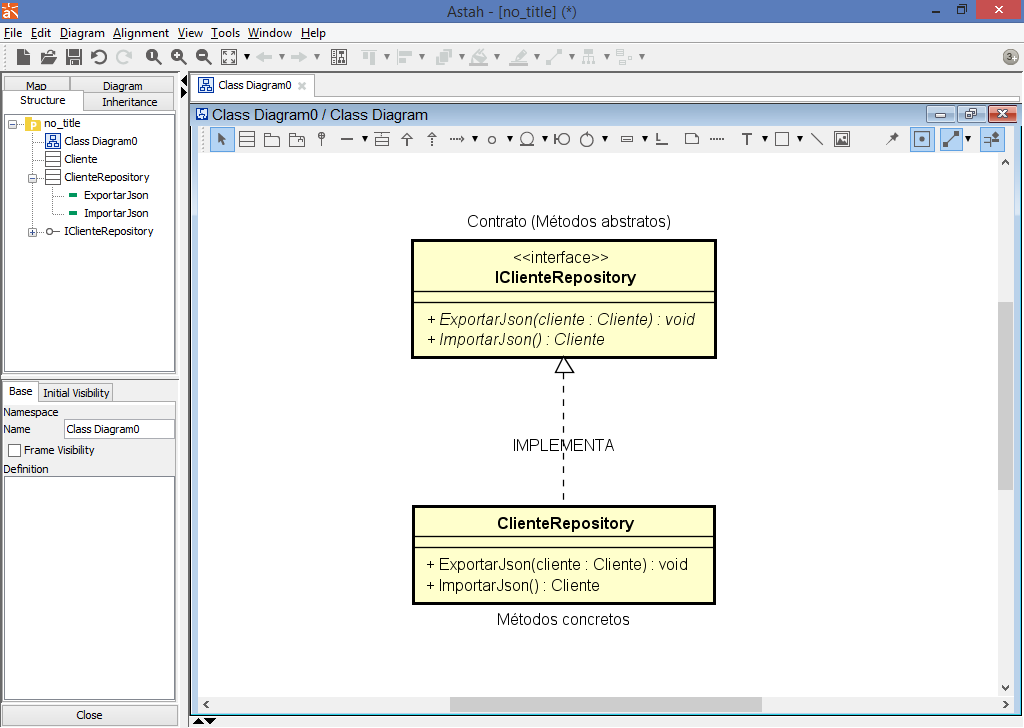
}

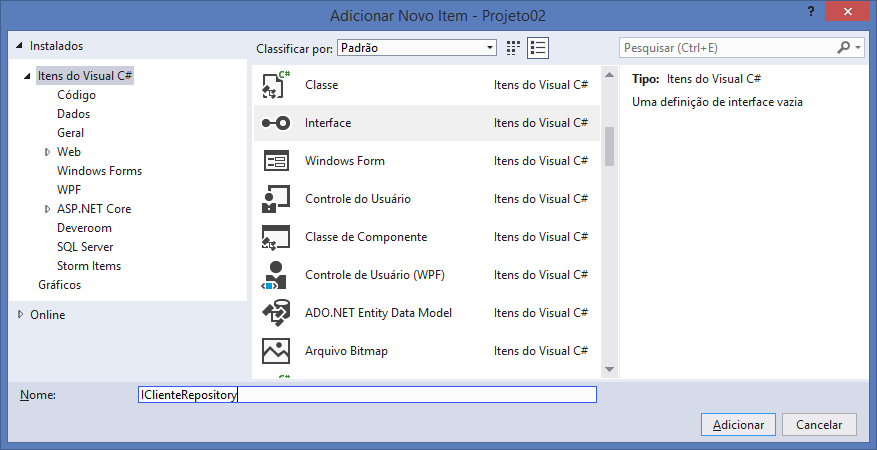
}

Interfaces (Contratos)

Uma interface é um programa composto somente de métodos abstratos, ou seja, métodos que não possuem corpo, apenas assinatura. Quando uma classe HERDA uma interface, a classe é obrigada a programar (IMPLEMENTAR) todos os métodos abstratos da interface.

Regra: Toda interface em C# tem o nome começando com a letra **I**





using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using Projeto02.Entities; //importando

namespace Projeto02.Contracts

{

//contrato

public interface IClienteRepository

{

//método abstrato (sem corpo, somente assinatura)

void ExportarJson(Cliente cliente);

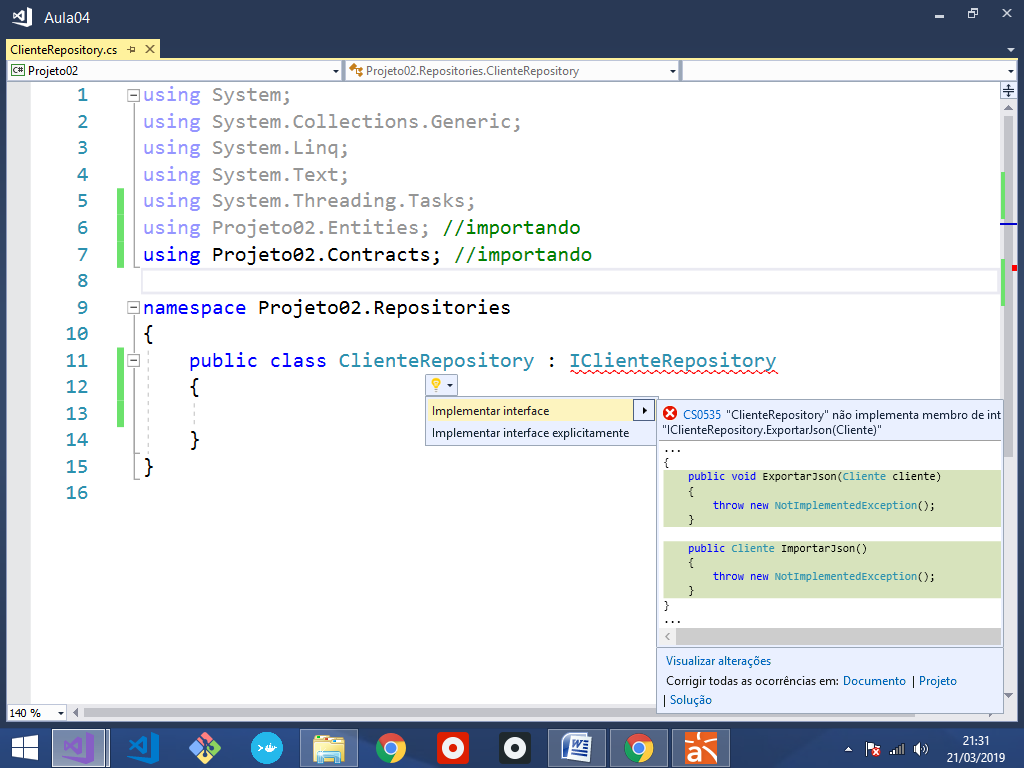
//método abstrato (sem corpo, somente assinatura)

Cliente ImportarJson();

}

}

**Vamos criar uma classe que IMPLEMENTE os métodos da interface:**



using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using Projeto02.Entities; //importando

using Projeto02.Contracts; //importando

namespace Projeto02.Repositories

{

public class ClienteRepository : IClienteRepository

{

public void ExportarJson(Cliente cliente)

{

throw new NotImplementedException();

}

public Cliente ImportarJson()

{

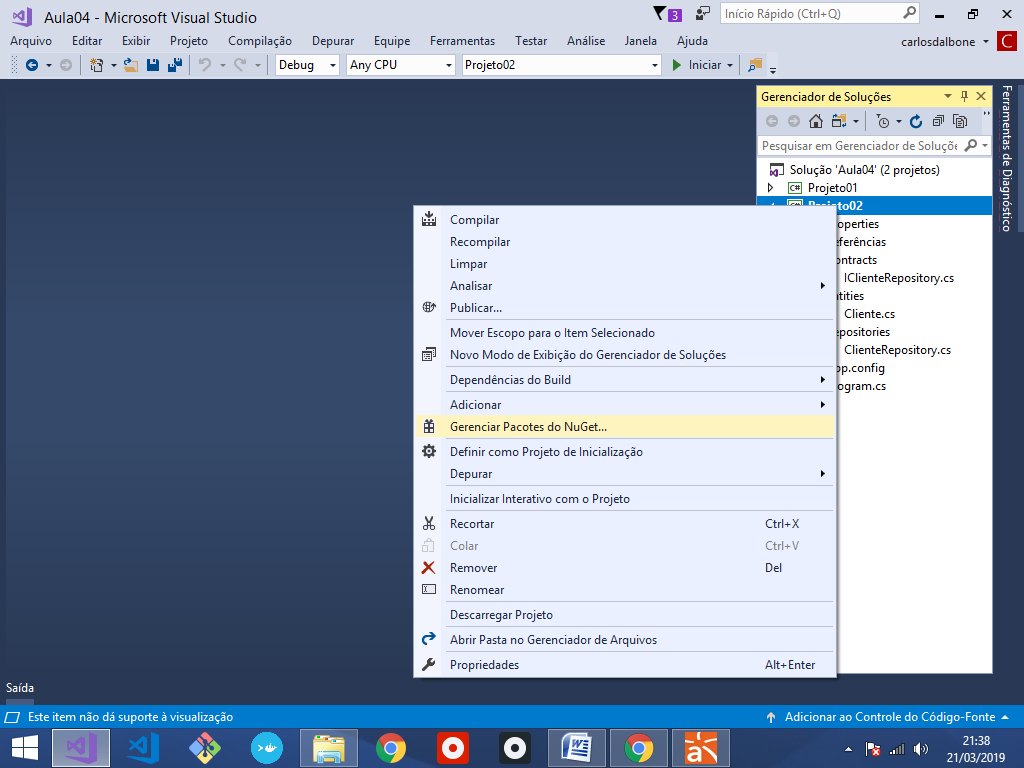
throw new NotImplementedException();

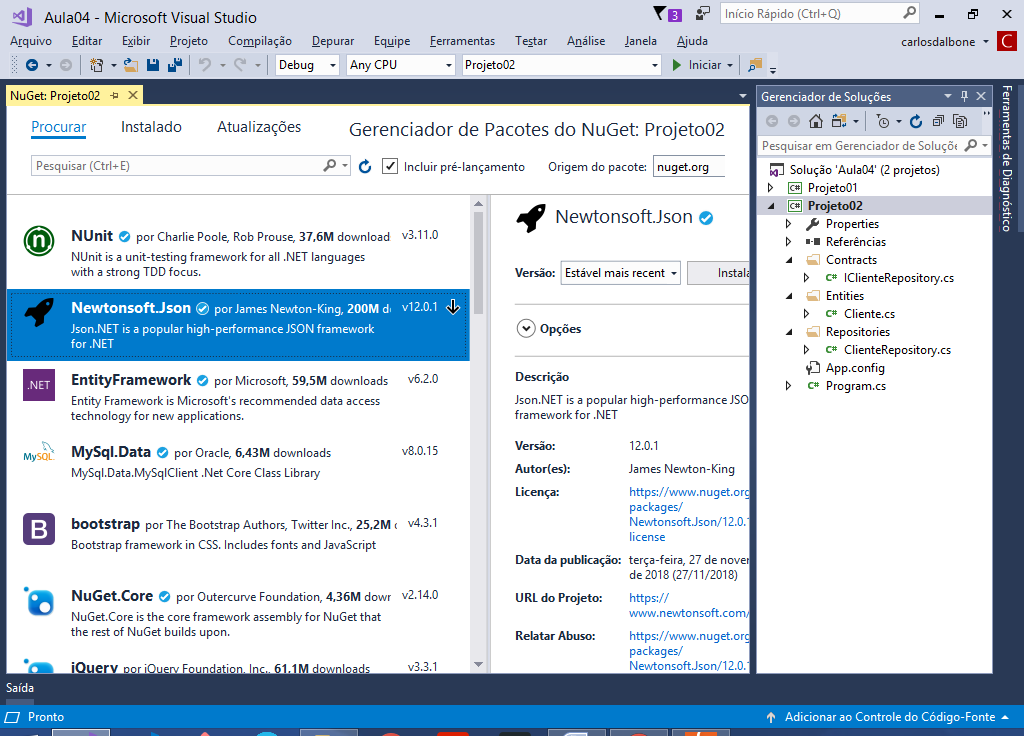
}

}

}

**Instalando a biblioteca para JSON:**





using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using Projeto02.Entities; //importando

using Projeto02.Contracts; //importando

using System.IO; //importando

using Newtonsoft.Json; //importando

namespace Projeto02.Repositories

{

public class ClienteRepository : IClienteRepository

{

public void ExportarJson(Cliente cliente)

{

using (StreamWriter streamWriter

= new StreamWriter("c:\\temp\\cliente.json"))

{

streamWriter.WriteLine

(JsonConvert.SerializeObject(cliente, Formatting.Indented));

}

}

public Cliente ImportarJson()

{

using (StreamReader streamReader

= new StreamReader("c:\\temp\\cliente.json"))

{

return JsonConvert.DeserializeObject<Cliente>

(streamReader.ReadToEnd());

}

}

}

}

**Testando:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using Projeto02.Entities; //importando..

using Projeto02.Repositories; //importando..

namespace Projeto02

{

class Program

{

static void Main(string[] args)

{

Cliente cliente = new Cliente(1, "Ana Paula", "anapaula@gmail.com");

try

{

ClienteRepository repository = new ClienteRepository();

repository.ExportarJson(cliente);

Console.WriteLine("\nJSON gravado com sucesso!\n");

Cliente resultado = repository.ImportarJson();

Console.WriteLine("Cliente: " + cliente.ToString());

}

catch(Exception e)

{

Console.WriteLine("Erro: " + e.Message);

}

Console.ReadKey();

}

}

}

**Executando:**

