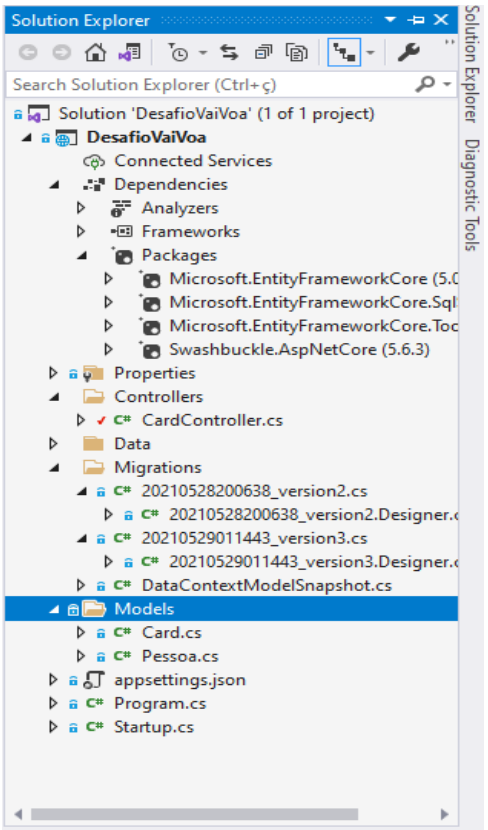
“Tutorial” sobre a criação de uma API para geração de número de cartão

A API é um conjunto de definições e protocolos usado no desenvolvimento e na integração de aplicações. Às vezes, as APIs são descritas como um contrato entre um provedor e um usuário de informações, estabelecendo o conteúdo exigido pelo consumidor (a chamada) e o conteúdo exigido pelo produtor (a resposta). Por exemplo, o design da API de um serviço meteorológico pode especificar que o usuário forneça um CEP e o produtor responda em duas partes, a primeira contendo a temperatura mais elevada e a segunda com a temperatura mais baixa. Pense nas APIs como um mediador entre os usuários ou clientes e os recursos ou serviços web que eles querem obter.

A intenção desse artigo era fornecer informações concretas e confiáveis, porém, comecei a estudar programação a pouco tempo, já na faculdade, e diante do desafio do programa VaiVoa, resolvi tentar.Confesso que foi uma experiência muito legal, frustrante, empolgante, uma mistura de sentimentos.O desafio proposto era a criação de uma Api REST cujo objetivo era fornecer um sistema para gerar números de cartão aleatórios, cartões esses atrelados a um email do solicitante. Enfim, o que quero dizer é que isso não chega a ser um tutorial, mas sim um registro do “Eu tentei”.

Aqui está O Explorador de Soluções do projeto



Criei duas classes, Card e Pessoa.

Classe Card

```
public class Card
{
    [Key]
    1 reference
    public int Id { get; set; }
    2 references
    public string Email { get; set; }
    2 references
    public string Number { get; set; }
    2 references
    public string Flag { get; set; }
    2 references
    public Pessoa Pessoa { get; set; }

    1 reference
    public Card()
    {
    }

    0 references
    public Card(int id, string email, string number, string flag, Pessoa pessoa)
    {
        Id = id;
        Email = email;
        Number = number;
        Flag = flag;
        Pessoa = pessoa;
    }
```

```
private readonly int[] Visa = { 4 };
public readonly string[] Flags = {"Visa"};

//SELECIONE A BANDEIRA DO CARTÃO:
//1 - VISA
1 reference
public string NewNumber(int optFlag)
{
    Random a = new Random();
    string numberCard = "";

    //Captura o número inicial do numero do cartão, identificando a bandeira.
    if (optFlag == 1)
    {
        numberCard = Visa[a.Next(Visa.Length)].ToString();
    }
    else
    {
        Console.WriteLine("Invalid option, select valid option!!");
    }


    while(numberCard.Length < 16) //Cria os 16 numeros que possuem um cartão de crédito
    {
        numberCard += a.Next(10);
    }
    return numberCard;
}
```

Classe Pessoa

```
public class Pessoa
{
    [Key]
    0 references
    public int IdPessoa  { get; set; }

    [Required(ErrorMessage = "Este campo é obrigatório")]
    [MaxLength(254, ErrorMessage = "Este campo deve conter no máximo 254 caracteres")]
    [MinLength(10, ErrorMessage = "Este campo deve conter no mínimo 10 caracteres")]
    2 references
    public string Email { get; set; }
    1 reference
    public ICollection<Card> ListCards { get; set; } = new List<Card>();


    1 reference
    public void AddCard(Card card)
    {
        ListCards.Add(card);
    }

}
```

Controllers

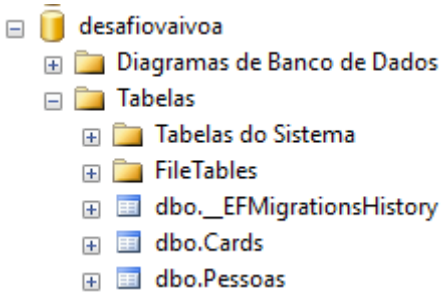
```
public class CardController : ControllerBase
{
    public readonly DataContext _context;

    [HttpPost]
    [Route("")]
    0 references
    public async Task<ActionResult<Card>> Post(
      [FromServices] DataContext context,
      [FromBody] Card model)

  {
      if (ModelState.IsValid)
      {
          string numCard = "";
          Card card = new Card();
          //Simulando a entrada do parâmentro email.
          Pessoa pessoa1 = new Pessoa();
          pessoa1.Email = "onediniz2 @gmail.com";
          int optFlag = 1;
          //Simulando a entrada do parâmetro opção optFlag = 1 , que é o cartão Visa.
          numCard = card.NewNumber(optFlag);
          card.Number = numCard;
          if(optFlag == 1)
          {
              card.Flag = "Visa";
          }
          card.Email = pessoa1.Email;
          card.Pessoa = pessoa1;
          pessoa1.AddCard(card);

          context.Cards.Add(model);
          await context.SaveChangesAsync();
          return model;
      }
      else
      {
          return BadRequest();
      }
```

Banco de Dados



Obrigado pelo desafio :)

Jorge Antonio Carneiro Diniz, 21.

Estudante: Análise e Desenvolvimento de Sistemas 3º Semestre

REFERÊNCIAS

REDHat <https://www.redhat.com/pt-br/topics/api/what-is-a-rest-api>acesso em:28,março de 2021.