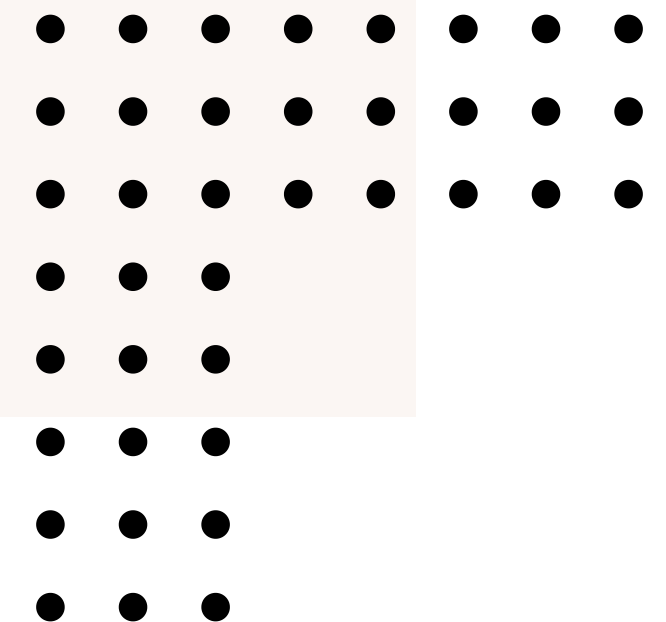
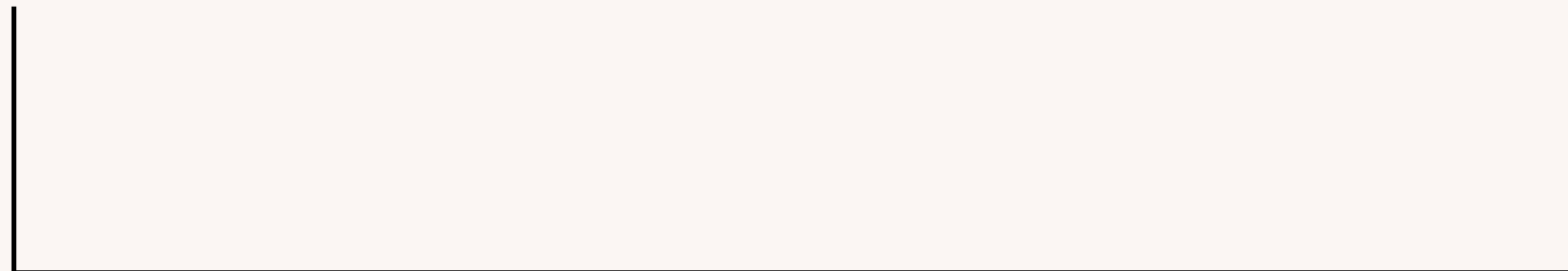
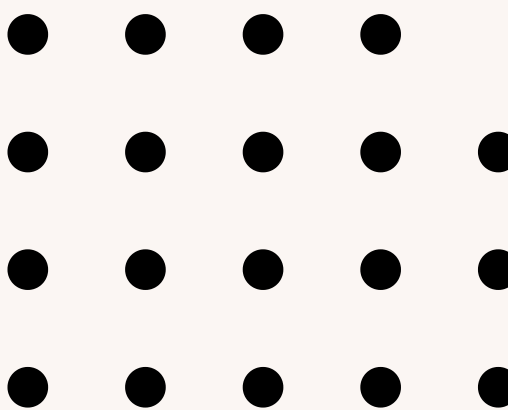


TREINAMENTO .NET/REST API



TÓPICOS DE ABORDAGEM

- Ecosystem .NET
- REST API e HTTP
- Começa a programar uma API de escolas
 - `dotnet new webapi`
- Parâmetros de rotas
- Parâmetros de query
- Atributos para validação



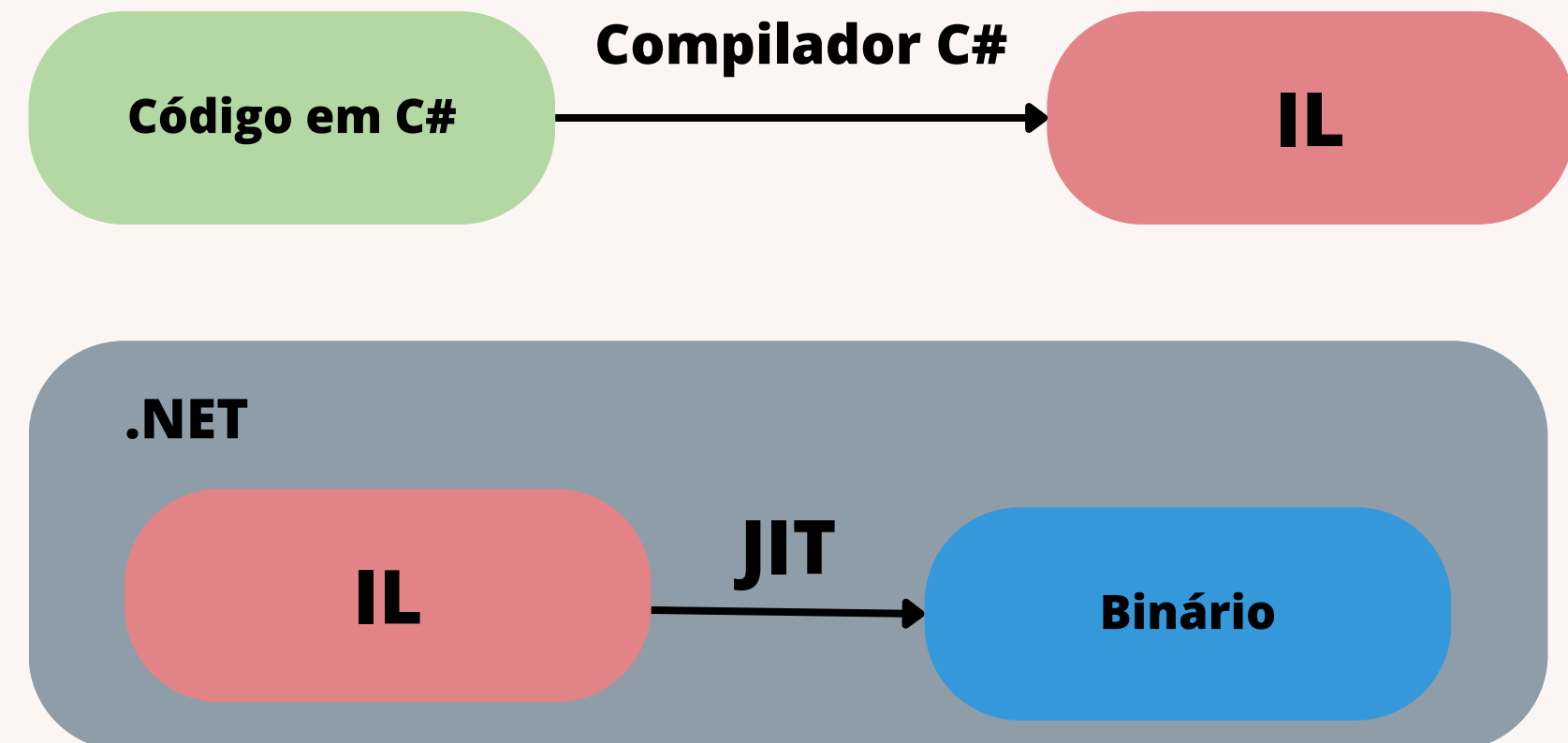
ARQUITETURA .NET

- Interoperabilidade;

MODELO NÃO GERENCIADO

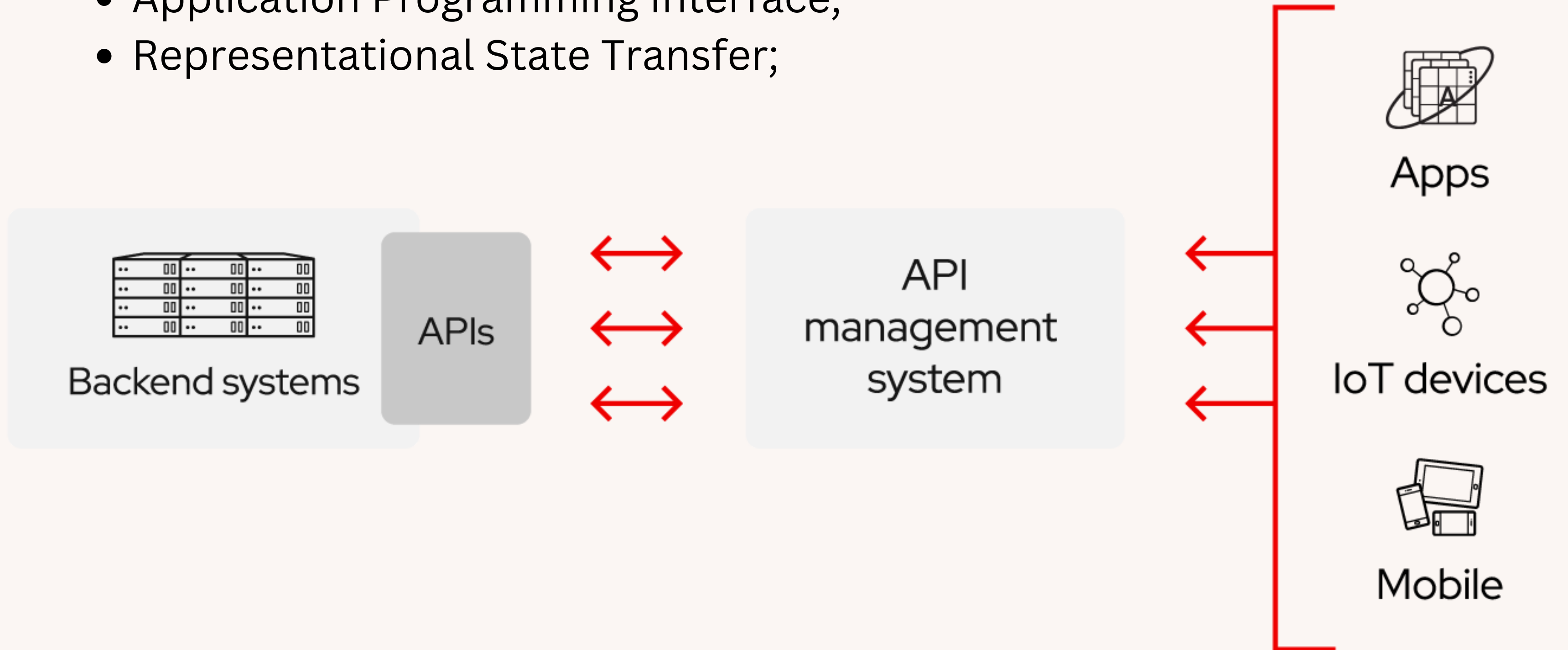


MODELO GERENCIADO



REST API

- Application Programming Interface;
- Representational State Transfer;



HTTP REPL

```
(Disconnected)> connect http://localhost:5004 --swagger /swagger/v1/swagger.json
Checking http://localhost:5004/swagger/v1/swagger.json... Found
Parsing... Successful
```

```
Using a base address of http://localhost:5004/
Using OpenAPI description at http://localhost:5004/swagger/v1/swagger.json
For detailed tool info, see https://aka.ms/http-repl-doc
```

```
http://localhost:5004/> ls
. []
Pizza [GET|POST]
WeatherForecast [GET]
```

```
http://localhost:5004/> cd Pizza
/Pizza [GET|POST]
```

```
http://localhost:5004/Pizza> ls
. [GET|POST]
.. []
{id} [GET|PUT|DELETE]
{num} []
```

```
http://localhost:5004/Pizza> ls -r
. [GET|POST]
.. []
{id} [GET|PUT|DELETE]
{num} []
{denom} [GET]
```

```
http://localhost:5004/Pizza> get 1/3
```

```
HTTP/1.1 200 OK
```

```
Content-Type: application/json; charset=utf-8
```

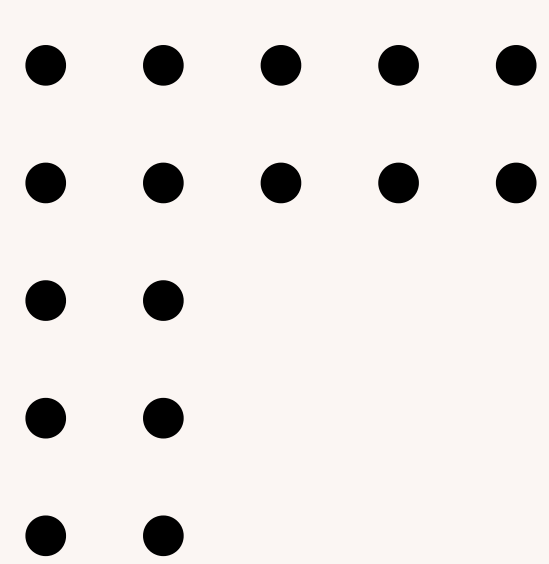
```
Date: Tue, 26 Sep 2023 15:46:10 GMT
```

```
Server: Kestrel
```

```
Transfer-Encoding: chunked
```

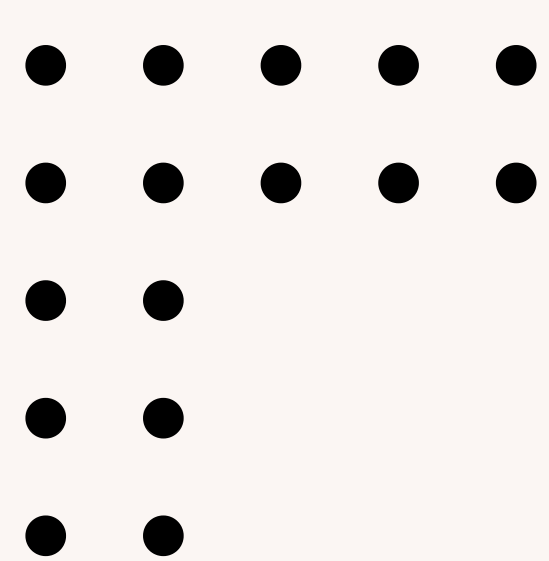
```
{
  "result": 0.33333334
}
```

```
dotnet tool install --global Microsoft.dotnet-httprepl
```



COISAS WEB

MDN | HTTP Messages



COISAS WEB

HTTP

Métodos/verbos HTTP:

GET

POST

PUT

PATCH

DELETE

OPTIONS

HEAD

COISAS WEB

REQUISIÇÃO HTTP

```
POST / HTTP/1.1
```

```
Host: localhost:8000
```

```
User-Agent: Mozilla/5.0 (Macintosh;... )... Firefox/51.0
```

```
Accept: text/html,application/xhtml+xml,..., */*;q=0.8
```

```
Accept-Language: en-US,en;q=0.5
```

```
Accept-Encoding: gzip, deflate
```

```
Connection: keep-alive
```

```
Upgrade-Insecure-Requests: 1
```

```
Content-Type: multipart/form-data; boundary=-12656974
```

```
Content-Length: 345
```

```
-12656974
```

```
(more data)
```

Request headers

General headers

Representation
headers

O que mais pode estar tem em uma requisição HTTP?

COISAS WEB

RESPOSTA HTTP

HTTP/1.1 200 OK

Access-Control-Allow-Origin: *

Connection: Keep-Alive

Content-Encoding: gzip

Content-Type: text/html; charset=utf-8

Date: Wed, 10 Aug 2016 13:17:18 GMT

Etag: "d9b3b803e9a0dc6f22e2f20a3e90f69c41f6b71b"

Keep-Alive: timeout=5, max=999

Last-Modified: Wed, 10 Aug 2016 05:38:31 GMT

Server: Apache

Set-Cookie: csrftoken=.....

Transfer-Encoding: chunked

Vary: Cookie, Accept-Encoding

X-Frame-Options: DENY

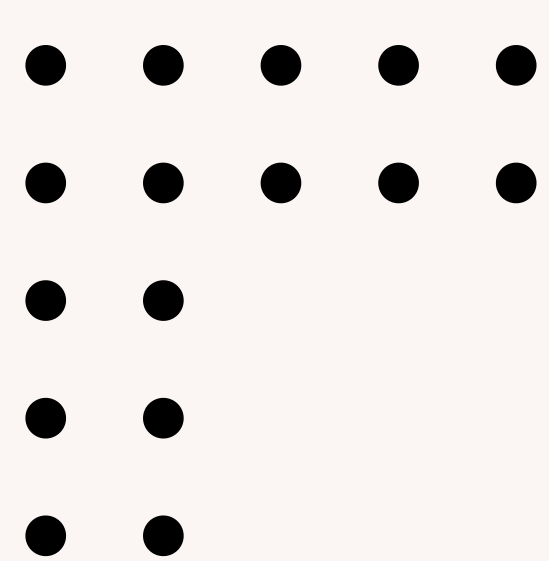
(body)

Response headers

Representation
headers

General headers

← O que vem no body de uma resposta HTTP?



COISAS WEB

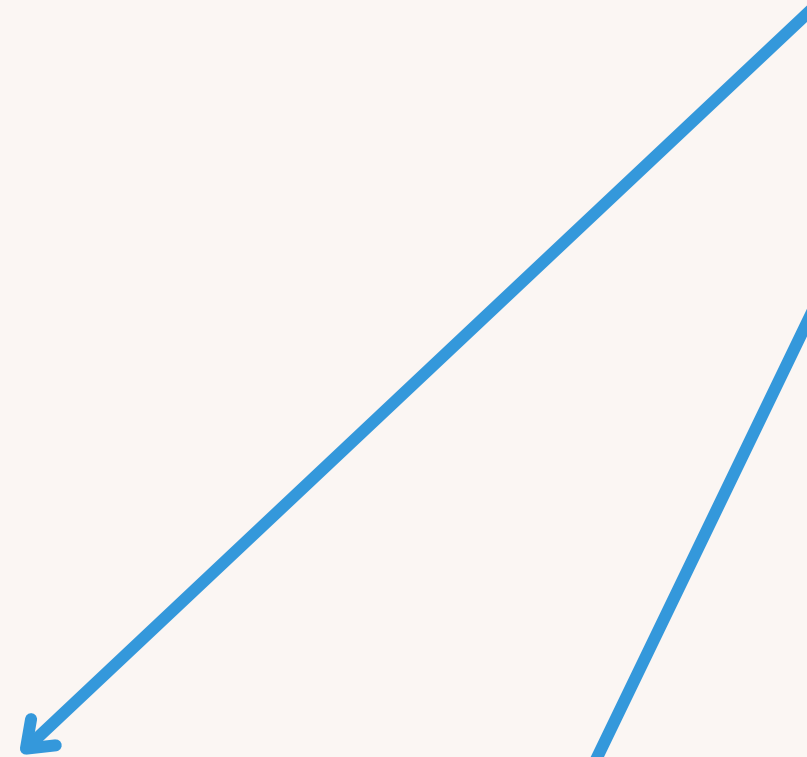
PARÂMETROS HTTP

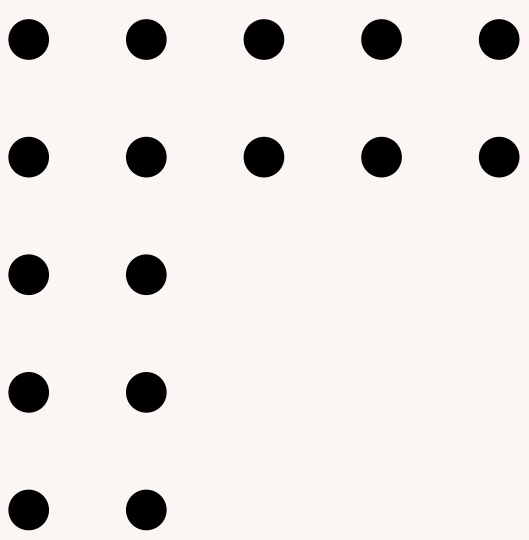
Exemplos de requisições:

GET /background.png
OPTIONS /anypage.html
GET /accounts/me
POST /user

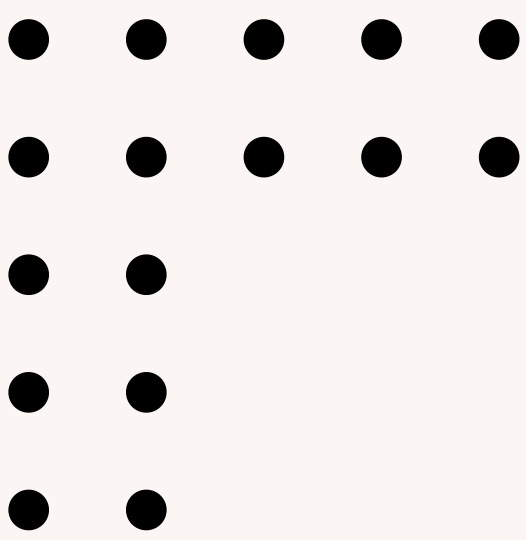
GET https://github.com/yudi-azvd/
GET /products?page=10&marca=samsung&preco=gte.100
HEAD /test.html?query=alibaba
GET https://www.youtube.com/watch?v=fcZXfoB2f70

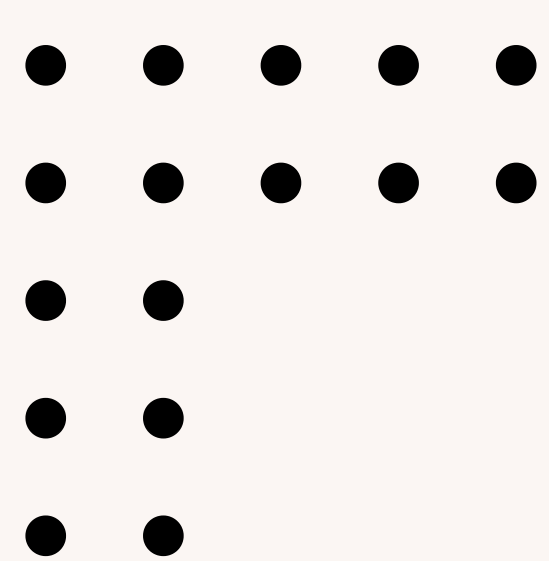
Parâmetros de rota
? query params



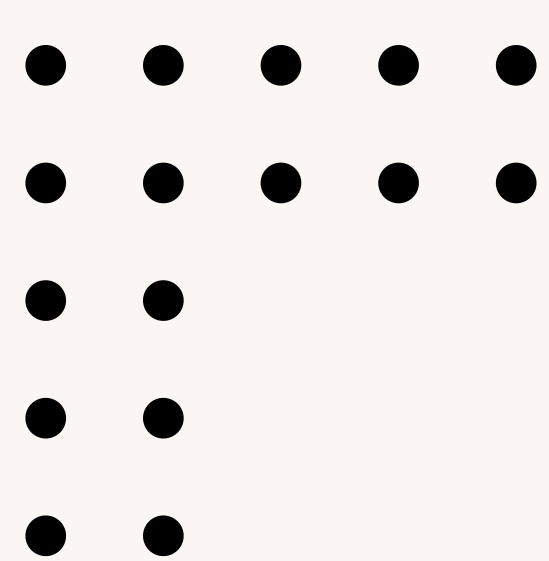


OBRIGADO



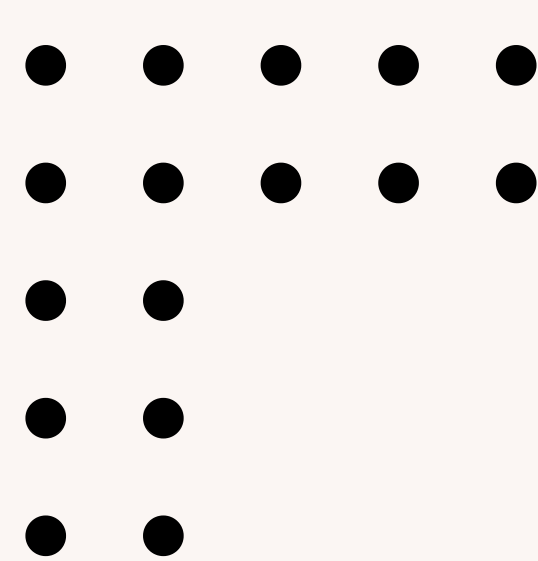


PADRÕES DE PROJETO

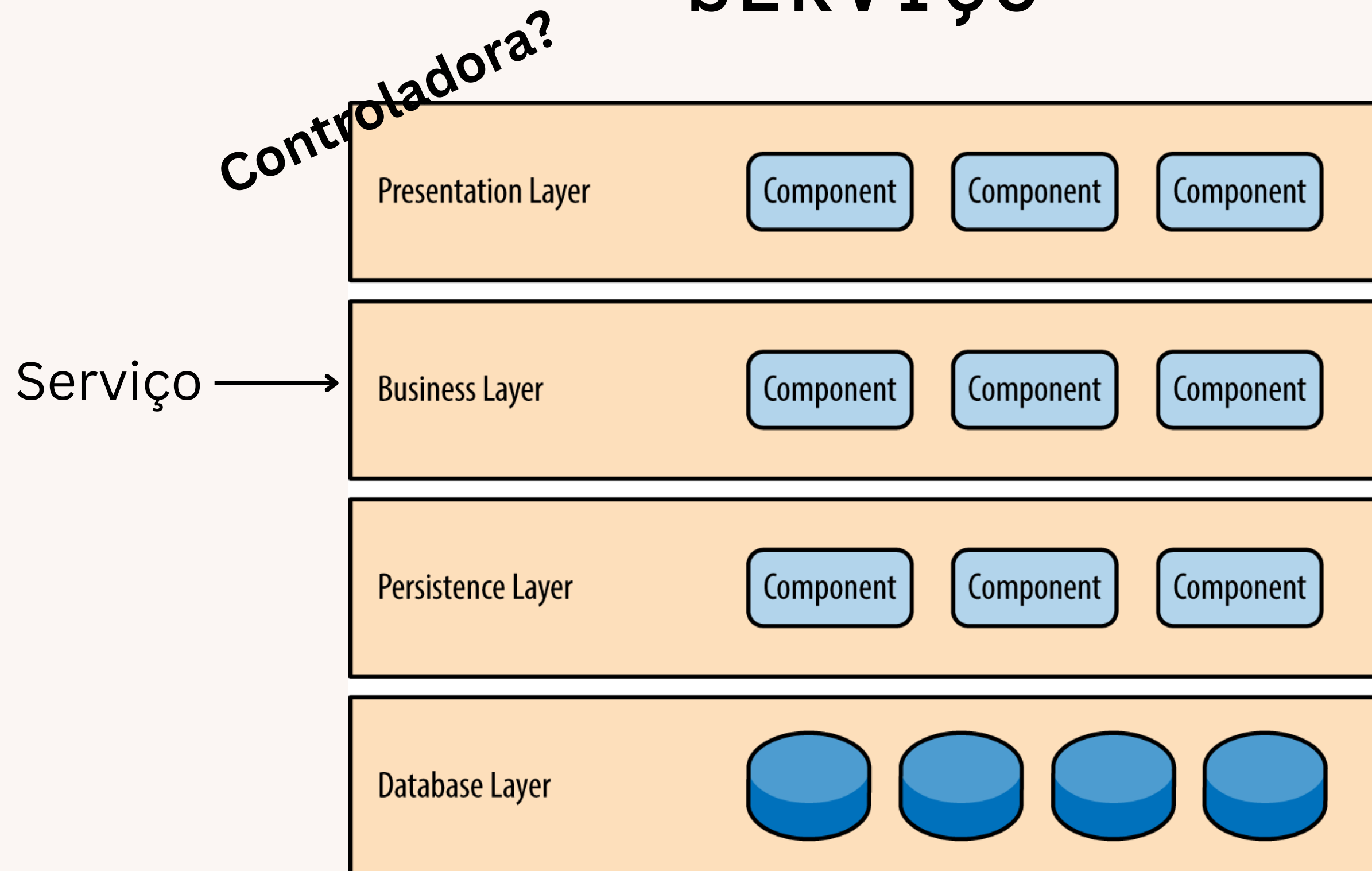


PADRÕES DE PROJETO: CONTROLADORA

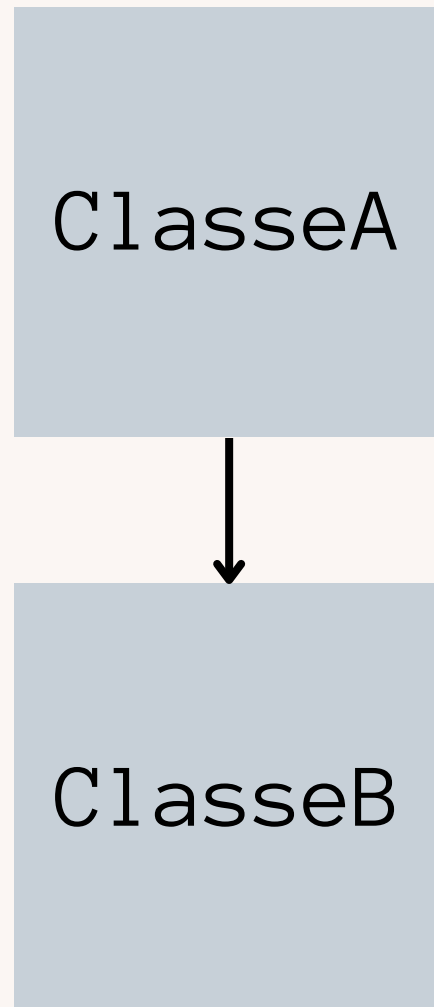
Responsável por responder requisições



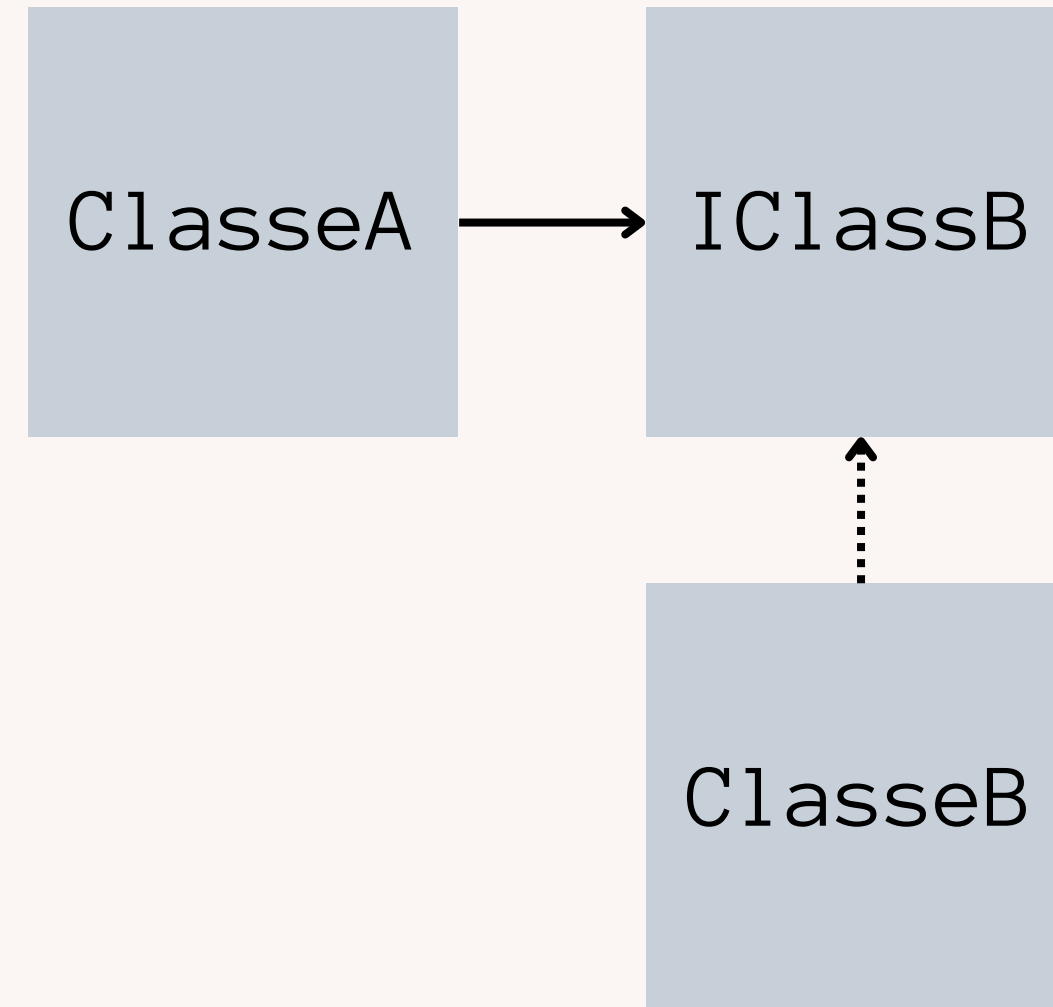
PADRÕES DE PROJETO: SERVIÇO



PADRÕES DE PROJETO: INVERSÃO DE DEPENDÊNCIA



Dependência tradicional



Inversão de dependência

PADRÕES DE PROJETO: INVERSÃO DE DEPENDÊNCIA

Menu.java src/Menu.java

```
import java.util.Scanner;

import controladoras.ControladoraAcesso;
import controladoras.ControladoraCadastro;
import controladoras.ControladoraPagamento;
import excecoes.OpcaoInvalidaException;

public class Menu {
    Scanner entradaDoTeclado = new Scanner(System.in);
    ControladoraCadastro controladoraCadastro = new ControladoraCadastro(entradaDoTeclado);
    ControladoraPagamento controladoraPagamento = new ControladoraPagamento(entradaDoTeclado);
    ControladoraAcesso controladoraAcesso = new ControladoraAcesso(entradaDoTeclado);

    public void executa() {
        int opcao = -1;

        do {
            try {
                mostrarOpcoes();
            } catch (OpcaoInvalidaException e) {
                System.out.println(e.getMessage());
            }
        } while (opcao != 0);
    }
}
```

Dependência tradicional

PADRÕES DE PROJETO: INVERSÃO DE DEPENDÊNCIA

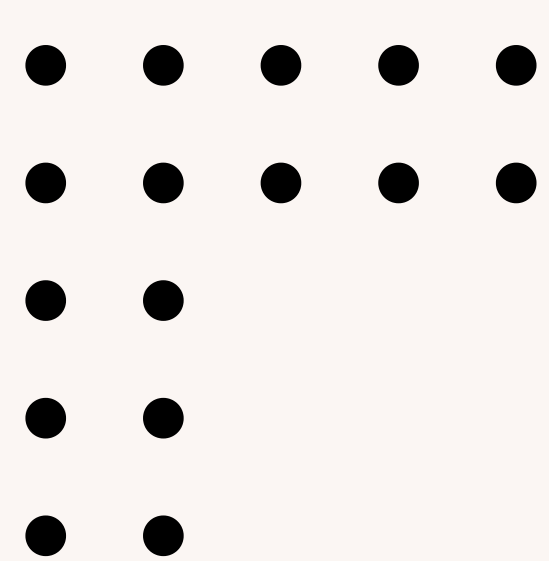
```
C# EscolaService.cs 2, M service/EscolaService.cs/ EscolaService

namespace service
{
    38 references
    public class EscolaService : IEscolaService
    {
        11 references
        private readonly IEscolaRepositorio escolaRepositorio;

        36 references
        public EscolaService(IEscolaRepositorio escolaRepositorio)
        {
            this.escolaRepositorio = escolaRepositorio;
        }
    }
}
```

Inversão de dependência

Dnit-EscolaService



PADRÕES DE PROJETO: REPOSITÓRIO

Persistência de dados

Arquivos?

NoSQL?

SQL?

ORM?

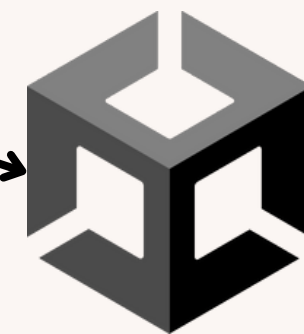
HISTÓRIA

- Cool;
- Lançado pela Microsoft em 2000;
- Fortes Princípios de OO;
- Influência de Java e C++;
- Cross-Platform e Open Source



HISTÓRIA

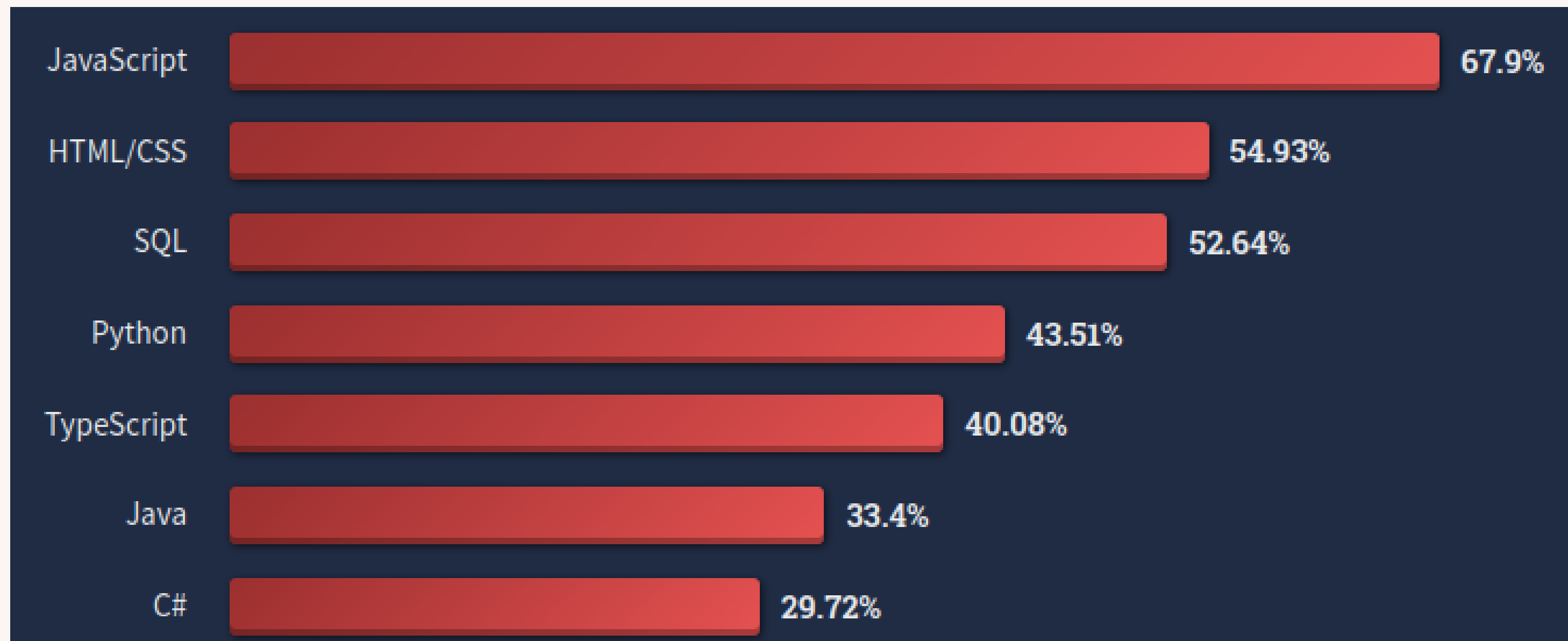
VERSATILIDADE



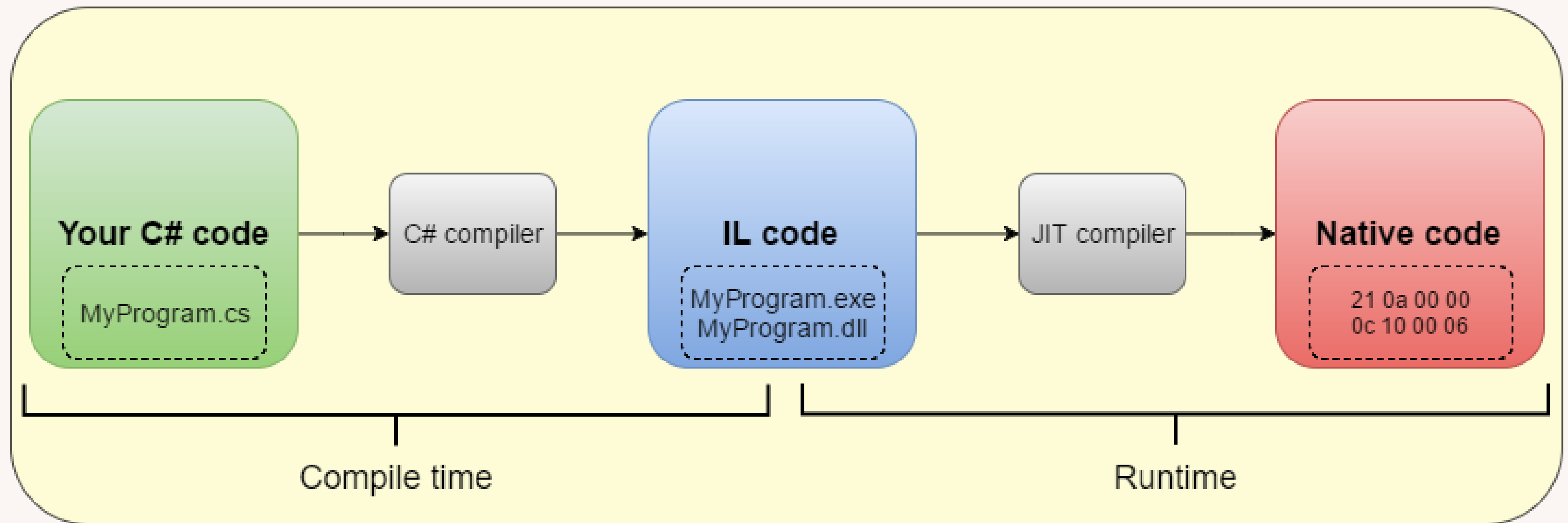
Unity

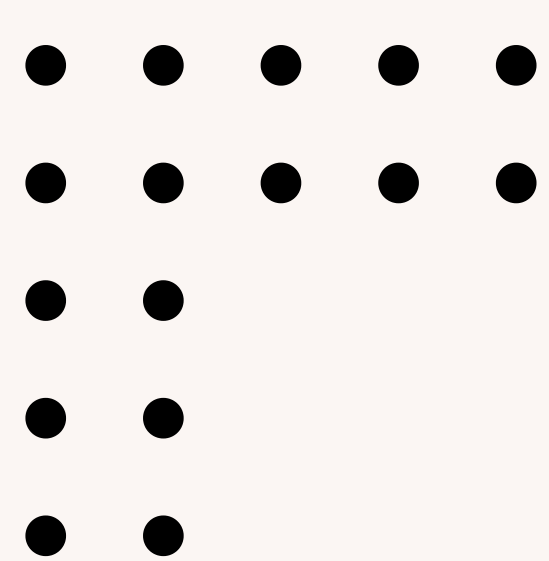


HISTÓRIA



PROCESSO DE COMPILAÇÃO





INSTALAÇÃO

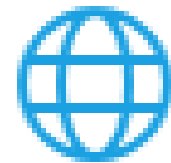
INSTALAÇÃO – WINDOWS

1



Baixe o
Visual Studio
Community.

2



ASP.NET e desenvolvimento Web



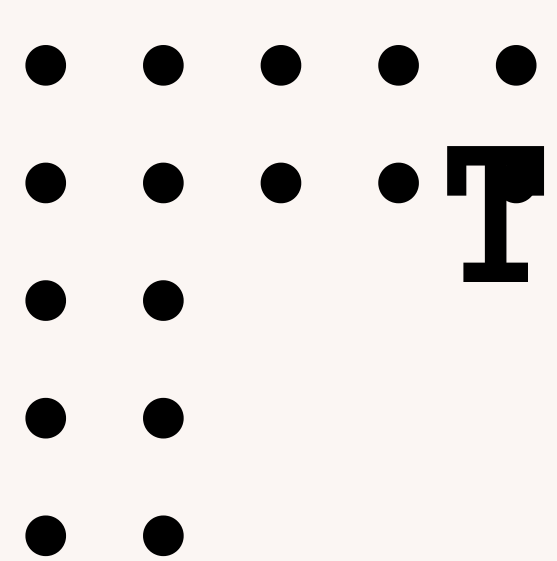
Crie aplicativos Web usando ASP.NET Core, ASP.NET,
HTML/JavaScript e Contêineres, incluindo suporte a Doc...

No instalador, marque a opção ASP.NET e
desenvolvimento Web

3



Instalar



TIPOS DE DADOS PRIMITIVOS

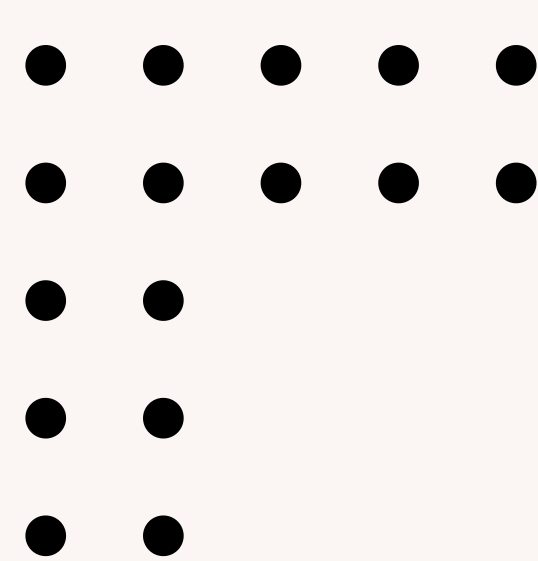
Tipo	Tamanho (bytes)	
int	4	int myNum = 5;
float	4	float myFloatNum = 5.99D;
double	8	double myDoubleNum = 5.99D;
char	1	char myLetter = 'D'
string	2 por caracter	string myText = "Hello"; string myText = \$"Soma 1 + 2 = {1 + 2}";
bool	1	bool myBool = true; bool myBool = false;



ARRAYS

```
string[] cars = new string[4] {"Volvo", "BMW", "Ford", "Mazda"};  
string[] cars = new string[] {"Volvo", "BMW", "Ford", "Mazda"};
```

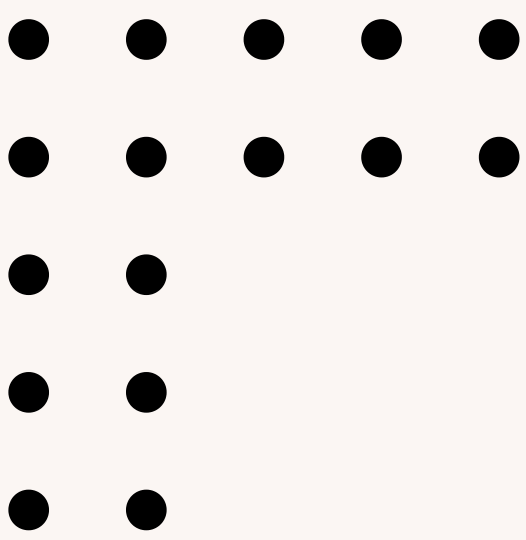
```
int[] myNum = {10, 20, 30, 40};  
int[] myNum = new int[]{10, 20, 30, 40};
```



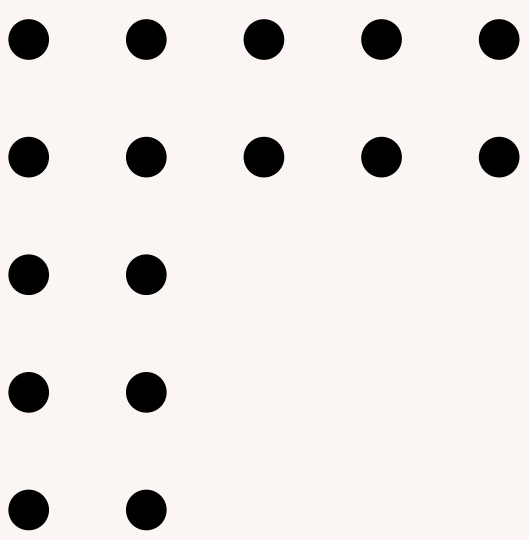
LISTAS

```
List<int> numeros = new List<int> { 1, 2, 3, 4, 5, 6 };  
var numeros = new List<int> { 1, 2, 3, 4, 5, 6 }
```

Adição de elementos	Append()	numeros.Append(10);
Maior elemento	Max()	var maior = numeros.Max();
Soma dos elementos	Sum()	var soma = numeros.Sum();
Conversão dos elementos	ConvertAll() Select()	var dobro = numeros.ConvertAll(x => x * 2); var triplo = numeros.Select(x => x * 3).ToList(); List<string> numerosString = numeros.ConvertAll(x => x.ToString());



ORIENTAÇÃO A OBJETOS



VEÍCULOS

