

AT - Desenvolvimento de Software Ágil e Escalável com Microsserviços



Alunos: Bernardo Correia de Miranda Alcântara Pereira

Curso: Engenharia da Computação

Professor: Rafael Bento Cruz

Matéria: Desenvolvimento de Software Ágil e Escalável com Microsserviços

Parte 1:

Questão 1 - Por fim, verificamos questões ligadas à arquitetura e à documentação. Neste contexto, responda:

1 - O que são microsserviços? Quando microsserviços são uma boa opção? Quando não são?

Respostas:

1 - Microsserviços são uma abordagem arquitetônica de desenvolvimento de software em que uma aplicação é construída como um conjunto de serviços independentes, cada um representando uma funcionalidade específica do sistema. Cada serviço é autônomo, possui sua própria base de código, banco de dados e pode ser implementado e escalado de forma independente. Esses serviços se comunicam entre si por meio de APIs.

Microsserviços são uma boa opção em vários cenários, incluindo:

Escalabilidade e Flexibilidade: Microsserviços permitem escalar partes específicas de uma aplicação de forma independente, facilitando a adaptação a picos de carga em áreas específicas do sistema.

Desenvolvimento Ágil: Equipes pequenas e ágeis podem trabalhar de forma mais eficiente em serviços menores e independentes. Cada equipe pode ser responsável por um ou mais microsserviços, permitindo um desenvolvimento paralelo e rápido.

Facilidade de Manutenção: Como os microsserviços são independentes, alterações em um serviço não afetam diretamente os outros. Isso facilita a manutenção e atualização contínua da aplicação.

Microsserviços podem não ser a melhor escolha em alguns contextos, como:

Complexidade Inicial: A introdução de microsserviços pode adicionar complexidade ao projeto, principalmente no início do desenvolvimento. Isso pode tornar o processo mais desafiador para equipes inexperientes.

Overhead de Comunicação: A comunicação entre microsserviços geralmente envolve o uso de APIs e chamadas de rede, o que pode introduzir um overhead significativo, especialmente em sistemas pequenos.

Custo de Infraestrutura: A gestão de vários microsserviços pode resultar em custos adicionais de infraestrutura, especialmente se cada serviço requer sua própria infraestrutura e recursos.

Coerência de Dados: Manter a consistência dos dados entre microsserviços pode ser desafiador. Transações distribuídas e garantias de consistência podem ser difíceis de implementar.

2 - Por que o método DDD de projeto de software ajuda na elaboração de microsserviços?

Resposta:

2 - O Domain-Driven Design (DDD) é benéfico para o design de microsserviços porque:

Modelagem de Domínio Rico: Facilita a compreensão do domínio do problema, auxiliando na identificação e divisão de microsserviços.

Delimitação de Contextos: O conceito de "Contextos Delimitados" ajuda a definir fronteiras claras para microsserviços, alinhando-se com a arquitetura.

Agrupamento Lógico de Funcionalidades: Identifica agregados, permitindo que microsserviços encapsulem entidades relacionadas.

Comunicação Clara entre Especialistas e Desenvolvedores: Promove uma linguagem comum, facilitando a comunicação e a compreensão mútua.

Evitar Anemia de Modelagem: Evita modelos de domínio simples sem comportamento, assegurando que os microsserviços encapsulem dados e comportamento.

Flexibilidade e Evolução Sustentável: Permite a evolução independente dos microsserviços ao longo do tempo.

Em resumo, o DDD fornece ferramentas e conceitos que ajudam na criação de microsserviços coesos, delimitados e que reflitam a complexidade do domínio do problema.

Quanto a comunicação entre microsserviços:

1 - O que é REST? O que é RESTful? Por que essa arquitetura é tão popular?

Respostas:

1 - REST é um estilo arquitetural utilizado na concepção de sistemas distribuídos, especialmente em serviços web. Ele é baseado em princípios que visam a simplicidade,

escalabilidade e desempenho. No contexto de serviços web, o REST utiliza operações HTTP padrão (GET, POST, PUT, DELETE) para realizar ações em recursos identificados por URLs.

Quando dizemos que uma aplicação ou serviço é RESTful, significa que ela adere aos princípios e restrições do estilo arquitetural REST. Isso inclui o uso correto dos métodos HTTP, a identificação clara de recursos através de URLs, a representação dos recursos em formatos como JSON ou XML, e a ausência de estado (statelessness) nas requisições.

A aceitação generalizada em APIs públicas e serviços web solidifica o REST como um padrão confiável e familiar para desenvolvedores, contribuindo para sua ampla popularidade na comunicação entre sistemas distribuídos.

2 - O que é um Message Broker?

Resposta:

2 - Um Message Broker é um componente de software que facilita a comunicação assíncrona entre sistemas distribuídos. Ele roteia mensagens, utiliza filas e tópicos, transforma mensagens se necessário, garante a entrega e proporciona desacoplamento entre emissores e receptores de mensagens. Exemplos incluem Apache Kafka, RabbitMQ e ActiveMQ, sendo comumente empregados em arquiteturas de microsserviços e sistemas distribuídos.

3 - Explique como é a implementação em Filas. RabbitMQ é uma boa tecnologia para essa implementação? Justifique.

Respostas:

3 - A implementação em filas, comumente conhecida como "Message Queues", é uma abordagem em que as mensagens são enviadas para uma fila intermediária antes de serem consumidas pelos destinatários. Essa técnica é valiosa para garantir o processamento assíncrono e o desacoplamento entre componentes de um sistema distribuído. O RabbitMQ é uma excelente tecnologia para implementar filas por várias razões:

Modelo de Filas: O RabbitMQ é construído com um modelo de filas robusto, oferecendo suporte eficiente para a criação, manipulação e gestão de filas de mensagens.

Protocolo AMQP: O RabbitMQ implementa o protocolo Advanced Message Queuing Protocol (AMQP), um padrão aberto e eficiente para a troca de mensagens. Isso facilita a interoperabilidade entre diferentes sistemas e linguagens de programação.

Desacoplamento: Ao utilizar filas, o RabbitMQ permite o desacoplamento entre produtores e consumidores de mensagens. Os produtores podem enviar mensagens para as filas sem se preocupar com quem ou como elas serão consumidas.

Escalabilidade: O RabbitMQ é escalável e pode lidar com um grande volume de mensagens. Sua arquitetura distribuída permite a expansão horizontal para atender a demandas crescentes.

Durabilidade e Persistência: O RabbitMQ suporta filas duráveis e persistência de mensagens, garantindo que as mensagens não sejam perdidas mesmo em caso de falhas no sistema.

Recursos Avançados: Oferece recursos avançados, como trocas de mensagens, roteamento flexível, filas de prioridade e mecanismos de confirmação, proporcionando flexibilidade na implementação de padrões de mensagens complexos.

Comunidade Ativa: O RabbitMQ possui uma comunidade ativa e é amplamente adotado na indústria, o que significa que há uma abundância de recursos, documentação e suporte disponíveis.

Portanto, o RabbitMQ é uma escolha sólida para a implementação de filas em sistemas distribuídos, proporcionando uma solução confiável, escalável e flexível para a troca eficiente de mensagens entre componentes.

4 - Descreva, com riqueza de detalhes, o padrão API Gateway.

Resposta:

4 - O padrão API Gateway é uma arquitetura que utiliza um componente centralizado para gerenciar e simplificar as interações entre clientes e serviços em uma aplicação distribuída. O API Gateway atua como um ponto de entrada único para todas as solicitações relacionadas à API, fornecendo uma camada de abstração sobre a complexidade subjacente dos serviços.

Principais Características:

Roteamento de Requisições: O API Gateway roteia as requisições dos clientes para os serviços apropriados com base em informações como URLs, cabeçalhos ou parâmetros.

Agregação de Dados: Ele pode reunir dados de múltiplos serviços em uma única resposta, reduzindo o número de chamadas necessárias pelo cliente e melhorando o desempenho.

Autenticação e Autorização: Gerencia a autenticação e a autorização, protegendo os serviços subjacentes contra acesso não autorizado e simplificando o processo de validação de tokens.

Balanceamento de Carga: Distribui o tráfego de maneira equitativa entre os serviços, facilitando a escalabilidade e o gerenciamento da carga.

Caching: Implementa caches para armazenar respostas frequentemente solicitadas, reduzindo a carga nos serviços e melhorando a latência.

Transformação de Dados: Pode realizar transformações nos dados para adequá-los ao formato desejado pelo cliente ou para agregar informações provenientes de diferentes fontes.

Monitoramento e Logging: Oferece recursos de monitoramento e logging, permitindo o rastreamento do tráfego, a identificação de possíveis problemas e a análise de desempenho.

Versionamento: Facilita o versionamento de APIs, permitindo a coexistência de diferentes versões e garantindo uma transição suave para os clientes.

Benefícios:

Simplicidade para os Clientes: Os clientes interagem apenas com o API Gateway, simplificando a integração e reduzindo a complexidade para os consumidores de serviços.

Desacoplamento: Os serviços subjacentes podem ser alterados ou substituídos sem afetar diretamente os clientes, promovendo o desacoplamento entre as camadas.

Segurança Centralizada: A centralização da autenticação e autorização no API Gateway simplifica a implementação e o gerenciamento de políticas de segurança.

Gerenciamento de Tráfego: Facilita o controle e a otimização do tráfego entre clientes e serviços, permitindo o monitoramento e a adaptação em tempo real.

Melhoria de Desempenho: A agregação de dados, o caching e o balanceamento de carga contribuem para melhorar a eficiência e o desempenho geral do sistema.

5 - O que é OpenAPI? Justifique com evidências.

Respostas:

OpenAPI é uma especificação para a descrição de APIs RESTful. Ela fornece uma maneira padronizada de documentar e definir os detalhes de uma API, incluindo informações sobre endpoints, parâmetros, respostas e outros aspectos relevantes. A especificação utiliza JSON ou YAML para descrever a estrutura da API de forma legível por máquina e humana.

Minhas justificativas:

Padrão Aberto e Neutro: A especificação OpenAPI é mantida pela OpenAPI Initiative, uma colaboração de diversas organizações, o que evidencia sua natureza de padrão aberto e neutro. A iniciativa conta com o suporte de empresas renomadas, como Google, Microsoft, IBM e outras.

Facilita a Colaboração: Ao adotar uma especificação comum como o OpenAPI, desenvolvedores, equipes e empresas podem colaborar de maneira eficaz na construção, documentação e consumo de APIs. A padronização reduz barreiras de entrada e promove a interoperabilidade.

Ferramentas de Geração de Documentação: O ecossistema OpenAPI é suportado por diversas ferramentas que facilitam a geração automática de documentação para APIs. Isso inclui ferramentas populares como Swagger UI e ReDoc. A capacidade de gerar documentação automaticamente a partir da especificação é uma evidência da utilidade do OpenAPI.

Suporte em Frameworks e Servidores: Muitos frameworks de desenvolvimento de APIs, como Express.js, Django REST Framework e Spring Boot, oferecem suporte direto ao OpenAPI. Além disso, servidores de API, como o Amazon API Gateway, também são compatíveis com a especificação, simplificando a integração e o desenvolvimento.

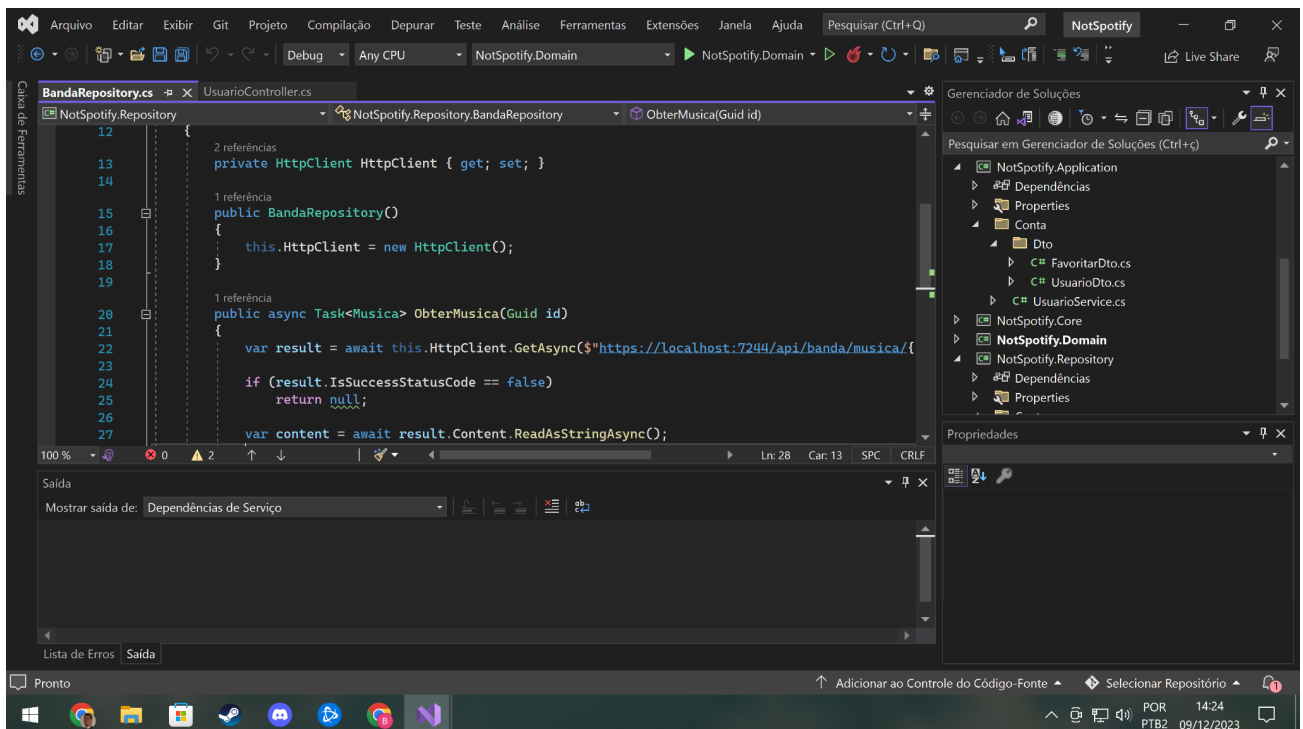
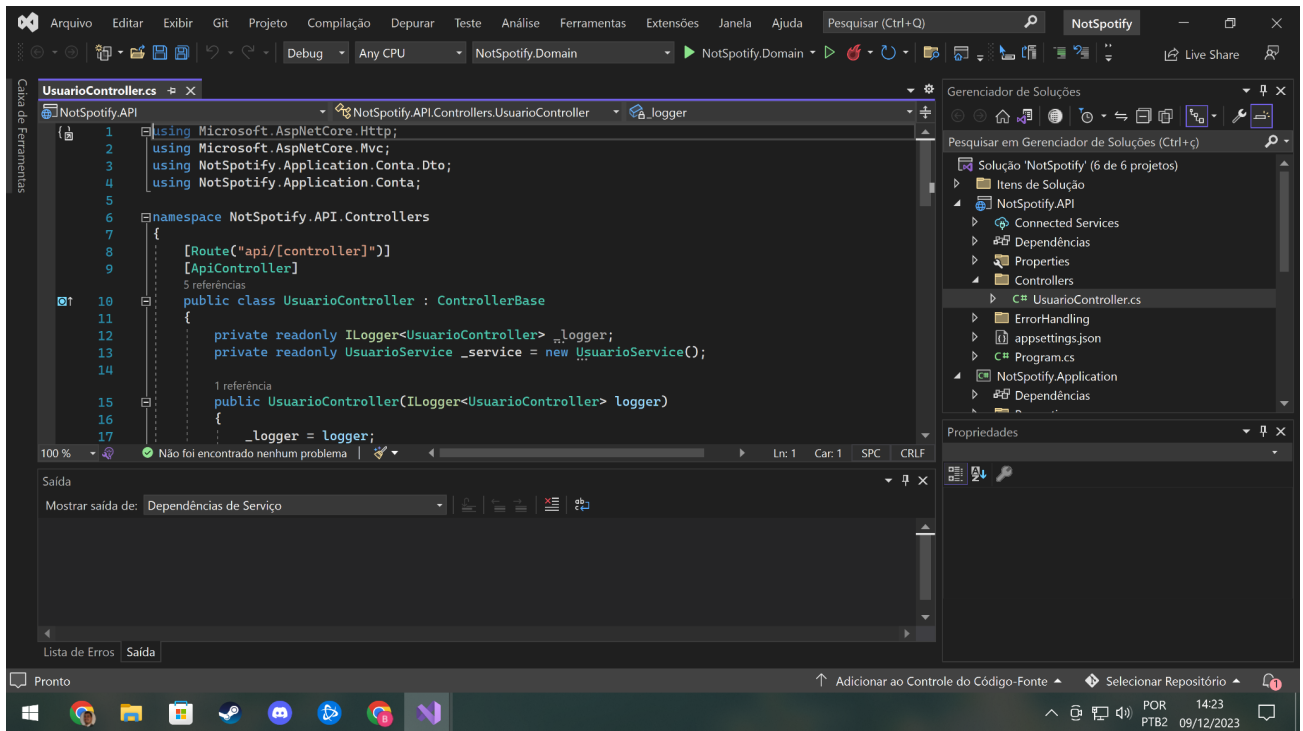
Validação e Teste Automático: O OpenAPI permite a validação automática de APIs, garantindo consistência e conformidade com a especificação. Ferramentas de teste, como Dredd e Postman, podem usar a especificação para automatizar testes e verificar se a implementação está de acordo com a documentação.

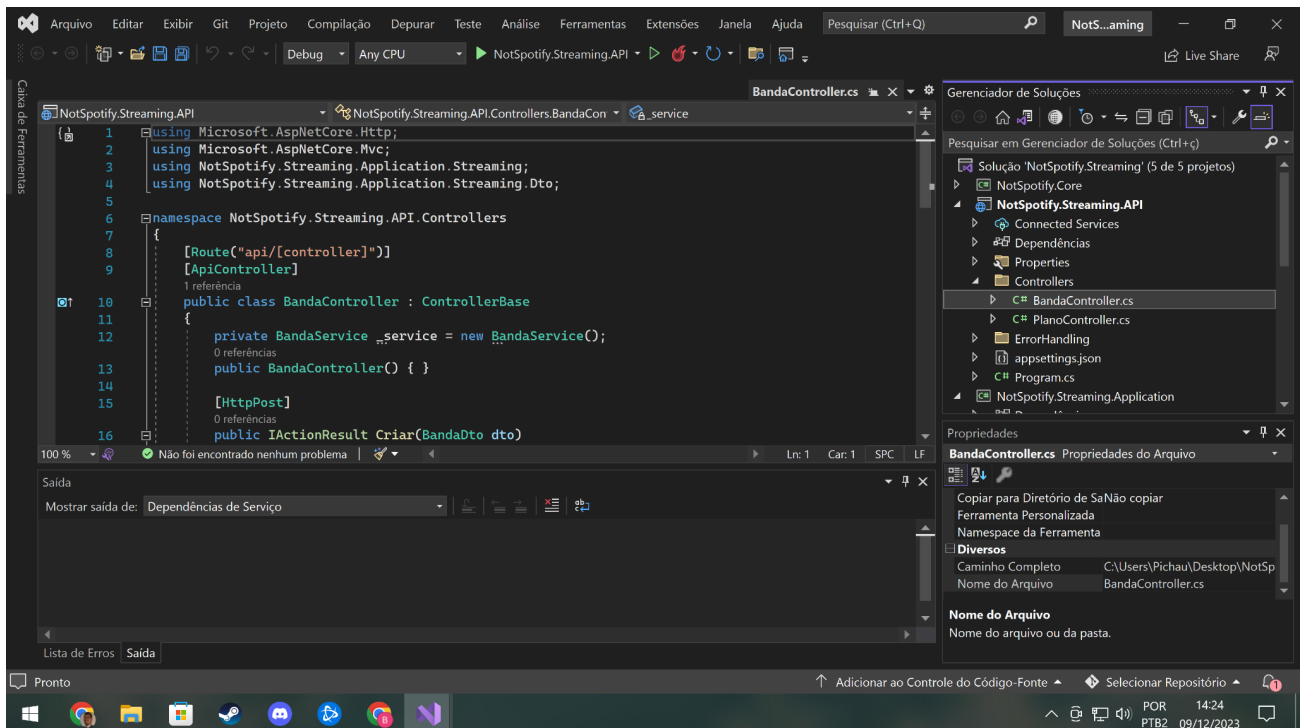
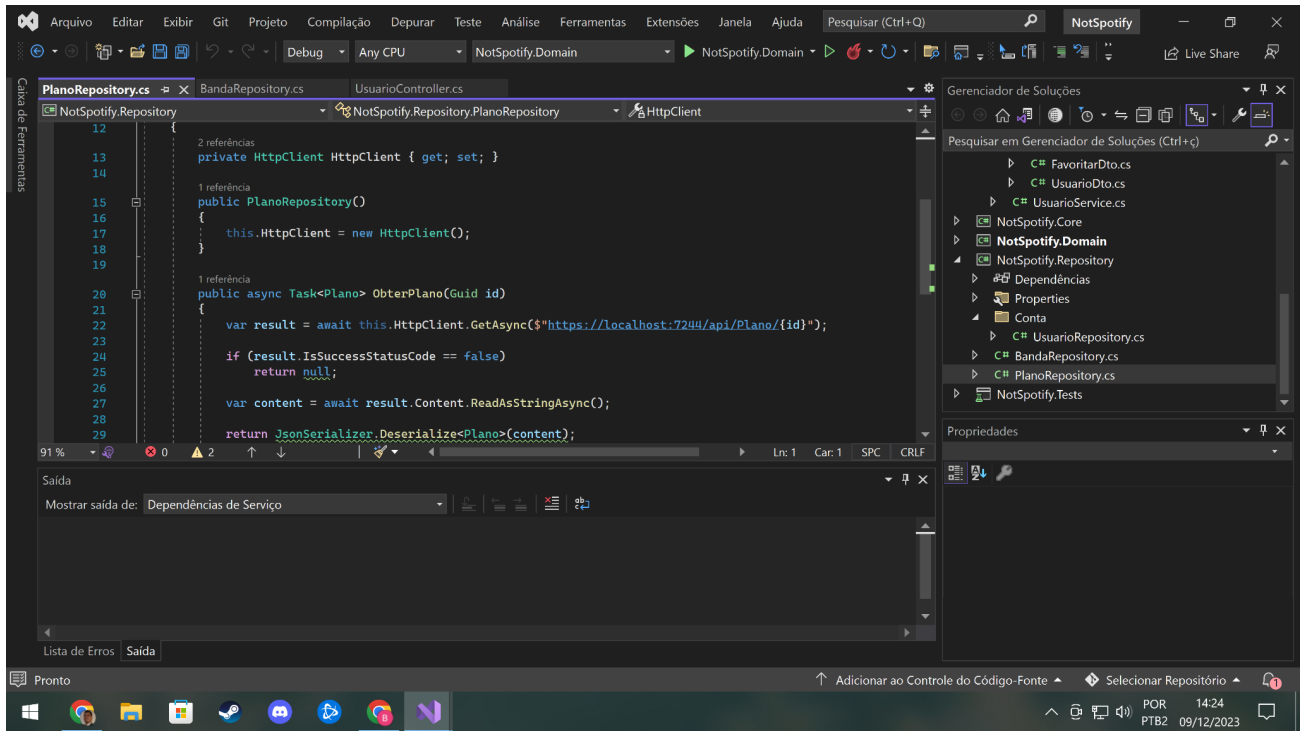
Parte 2:

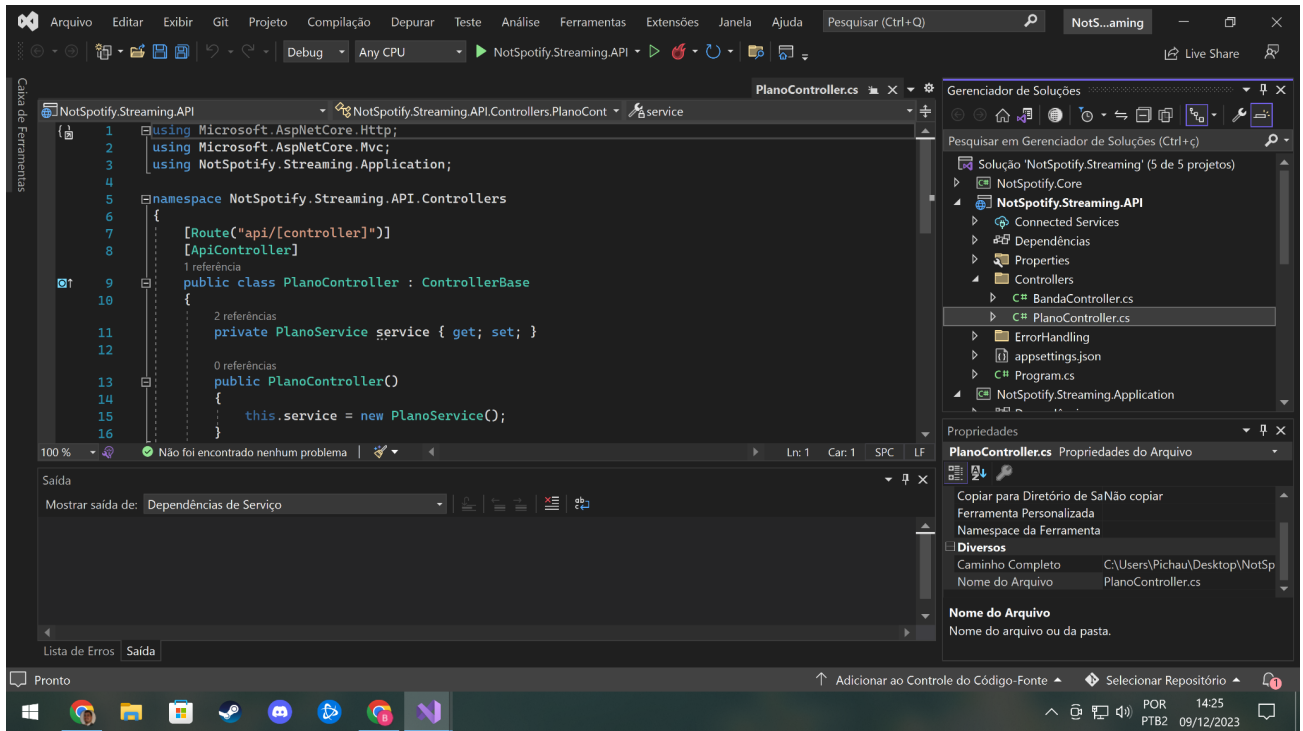
Imagens e links relacionados ao código do AT.

Link do github: <https://github.com/BerPereira/AT-Infnet-Microservices>

Evidências em imagens do funcionamento de ambas as APIs, comprovando a comunicação entre os microsserviços, evidências de que os dois projetos podem ter as APIs consumidas e evidências da publicação das APIs dos dois projetos:







Banda

POST/api/Banda

Parameters

No parameters

Request body

application/json

```
{  "nome": "Iron Maiden",  "genero": "Metal",  "descricao": "Banda inglesa",  "albums": [    {      "nome": "Fear of the Dark",      "dtLancamento": "11 de maio de 1992",      "musicas": [        {          "nome": "Fear of the Dark",          "duracao": 719        },        {          "nome": "Be Quick or Be Dead",          "duracao": 324        }      ]    }  ]}
```

ExecuteClear

Responses

Curl

```
curl -X 'POST' \  'https://localhost:7244/api/Banda' \  -H 'accept: */*' \  -H 'Content-Type: application/json' \  -d '{    "nome": "Iron Maiden",    "genero": "Metal",    "descricao": "Banda inglesa",    "albums": [      {        "nome": "Fear of the Dark",        "dtLancamento": "11 de maio de 1992",        "musicas": [          {            "nome": "Fear of the Dark",            "duracao": 719          },          {            "nome": "Be Quick or Be Dead",            "duracao": 324          }        ]      }    ]  }'
```

Request URL

https://localhost:7244/api/Banda

Server response

CodeDetails

201

undocumented

Response body

```
{  "id": "bcf5bd58-7a18-4898-891f-752504fd1db",  "nome": "Iron Maiden",  "genero": "Metal",  "descricao": "Banda inglesa",  "albums": [    {      "id": "00000000-0000-0000-0000-000000000000",      "nome": "Fear of the Dark",      "dtLancamento": "11 de maio de 1992",      "musicas": [        {          "id": "00000000-0000-0000-0000-000000000000",          "nome": "Fear of the Dark",          "duracao": 719        },        {          "id": "00000000-0000-0000-0000-000000000000",          "nome": "Be Quick or Be Dead",          "duracao": 324        }      ]    }  ]}
```

Download

Response headers

```
content-type: application/json; charset=utf-8date: Sat, 09 Dec 2023 18:38:47 GMTlocation: /banda/bcf5bd58-7a18-4898-891f-752504fd1dbserver: Kestrel
```

Responses

CodeDescriptionLinks

200

Success

No links

GET

/api/Banda/{id}

Parameters

Cancel

Name	Description
id required string(\$uuid) (path)	<div>bcf5bd58-7a18-4898-891f-752504fd1d0b</div>

ExecuteClear

Responses

Curl

```
curl -X 'GET' \
'https://localhost:7244/api/Banda/bcf5bd58-7a18-4898-891f-752504fd1d0b' \
-H 'accept: */*'
```

Request URL

```
https://localhost:7244/api/Banda/bcf5bd58-7a18-4898-891f-752504fd1d0b
```

Server response

Code

Details

200

Response body

```
{
  "id": "bcf5bd58-7a18-4898-891f-752504fd1d0b",
  "nome": "Iron Maiden",
  "genero": "Metal",
  "descricao": "Banda inglesa",
  "albums": [
    {
      "id": "6870eee2-ed1f-4dc3-aec8-e5a779d4c97c",
      "nome": "Fear of the Dark",
      "distanamento": "11 de maio de 1992",
      "musicas": [
        {
          "id": "3b5f286a-0e43-4ce9-ae95-5f651cd74f5b",
          "nome": "Fear of the Dark",
          "duracao": 719
        },
        {
          "id": "d6f3450c-6637-4112-93d0-a329533e240e",
          "nome": "Be Quick or Be Dead",
          "duracao": 324
        }
      ]
    }
  ]
}
```

Download

Response headers

```
content-type: application/json; charset=utf-8
date: Sat,09 Dec 2023 18:40:10 GMT
server: Kestrel
```

Responses

Code	Description	Links
200	Success	No links

Usuario

POST

/api/Usuario

Parameters

No parameters

Request body

application/json

```
{  "nome": "Bernardo",  "sobrenome": "Pereira",  "cpf": "64577195038",  "planoId": "80044595-D4A6-4E1A-9F09-DA892205C71C",  "cartao": {    "numero": "5248581002684983",    "limite": 2000,    "ativo": true  },  "playlists": []}
```

Execute

Clear

Responses

Curl

```
curl -X 'POST' \  'https://localhost:7084/api/Usuario' \  -H 'accept: */*' \  -H 'Content-Type: application/json' \  -d '{  "nome": "Bernardo",  "sobrenome": "Pereira",  "cpf": "64577195038",  "planoId": "80044595-D4A6-4E1A-9F09-DA892205C71C",  "cartao": {    "numero": "5248581002684983",    "limite": 2000,    "ativo": true  },  "playlists": []}'
```

Request URL

https://localhost:7084/api/Usuario

Server response

Code	Details
201	<div><div>undocumented</div><div>Response body</div><div><pre>{ "id": "da06ba62-d2ec-4fb1-a02a-5491dacc29cc", "nome": "Bernardo", "sobrenome": "Pereira", "cpf": "64577195038", "planoId": "80044595-d4a6-4e1a-9f09-dab92205c71c", "cartao": { "numero": "5248581002684983", "limite": 2000, "ativo": true }, "playlists": []}</pre></div><div>Download</div></div>

Response headers

```
content-type: application/json; charset=utf-8
date: Sat,09 Dec 2023 15:44:13 GMT
location: /usuario/da06ba62-d2ec-4fb1-a02a-5491dacc29cc
server: Kestrel
```

Responses

Code	Description	Links
200	Success	No links

GET

/api/Usuario/{id}

Parameters

Cancel

Name	Description
<div><div>id required</div><div>string(\$uid)</div><div>(path)</div></div>	<div>da06ba62-d2ec-4fb1-a02a-5491dacc29cc</div>

Execute

Clear

Responses

Curl

```
curl -X 'GET' \
'https://localhost:7884/api/Usuario/da06ba62-d2ec-4fb1-a02a-5491dacc29cc' \
-H 'accept: */*'
```

Request URL

```
https://localhost:7884/api/Usuario/da06ba62-d2ec-4fb1-a02a-5491dacc29cc
```

Server response

Code	Details
200	<div><div>Response body</div><div><pre>{ "id": "da06ba62-d2ec-4fb1-a02a-5491dacc29cc", "nome": "Bernardo", "sobrenome": "Pereira", "cpf": "64577195038-", "planoId": "00000000-0000-0000-0000-000000000000", "cartao": { "numero": "XXXX-XXXX-XXXX-XX", "limite": 2000, "ativo": true }, "playlists": [{ "id": "240863ee-ec9-4fc4-a4c0-f426018159ae", "nome": "Favoritas", "publica": false, "musicas": [] }] }</pre></div><div>Download</div></div>

Response headers

```
content-type: application/json; charset=utf-8
date: Sat,09 Dec 2023 18:44:59 GMT
server: Kestrel
```

Responses

Code	Description	Links
200	Success	No links

POST /api/Usuario/{id}/favoritar

Parameters

Cancel

Reset

Name	Description
id * required string(\$uuid) (path)	da06ba62-d2ec-4fb1-a02a-5491dacc29cc

Request body

application/json

```
{  "idMusica": "d6f3450c-6637-4112-93d0-a329533e240e"}
```

Execute

Clear

Responses

Curl

curl -X 'POST' \ 'https://localhost:7084/api/Usuario/da06ba62-d2ec-4fb1-a02a-5491dacc29cc/favoritar' \ -H 'accept: */*' \ -H 'Content-Type: application/json' \ -d '{ "idMusica": "d6f3450c-6637-4112-93d0-a329533e240e" }'

Request URL

https://localhost:7084/api/Usuario/da06ba62-d2ec-4fb1-a02a-5491dacc29cc/favoritar

Server response

Code	Details
200	<div>Response headers<div>content-length: 0 date: Sat, 09 Dec 2023 18:47:11 GMT server: Kestrel</div></div>

Responses

Code	Description	Links
200	Success	No links

GET

/api/Usuario/{id}

⌵

Parameters

Cancel

Name	Description
id * required string(\$uuid) (path)	da06ba62-d2ec-4fb1-a02a-5491dacc29cc

Execute

Clear

Responses

Curl

```
curl -X 'GET' \
  'https://localhost:7084/api/Usuario/da06ba62-d2ec-4fb1-a02a-5491dacc29cc' \
  -H 'accept: */*'
```

Request URL

https://localhost:7084/api/Usuario/da06ba62-d2ec-4fb1-a02a-5491dacc29cc

Server response

Code	Details
200	<div><div>Response body</div><div><pre>{ "id": "da06ba62-d2ec-4fb1-a02a-5491dacc29cc", "nome": "Bernardo", "sobrenome": "Pereira", "cpf": "64577195038-", "planoId": "00000000-0000-0000-0000-000000000000", "cartao": { "numero": "xxxx-xxxx-xxxx-xx", "limite": 2000, "ativo": true }, "playlists": [{ "id": "3d0863ee-ecc9-4fc4-a4c0-f426018159ae", "nome": "Favoritas", "publica": false, "musicas": [{ "id": "3b5f286a-0e43-4ce9-ae95-5f651cd74f5b", "nome": "Fear of the Dark", "duracao": 719 }, { "id": "d6f3450c-6637-4112-93d0-a329533e240e", "nome": "Be Quick or Be Dead", "duracao": 324 }] }] }</pre></div><div><div>Download</div></div></div>

Response headers

```
content-type: application/json; charset=utf-8
date: Sat, 09 Dec 2023 18:47:16 GMT
server: Kestrel
```

Responses

Code	Description	Links
200	Success	No links

GET

/api/Banda/musica/{id}

Parameters

Cancel

Name	Description
id * required string(\$uuid) (path)	<div>3b5f286a-0e43-4ce9-ae95-5f651cd74f5b</div>

Execute

Clear

Responses

Curl

```
curl -X 'GET' \
'https://localhost:7244/api/Banda/musica/3b5f286a-0e43-4ce9-ae95-5f651cd74f5b' \
-H 'accept: */*'
```

Request URL

https://localhost:7244/api/Banda/musica/3b5f286a-0e43-4ce9-ae95-5f651cd74f5b

Server response

Code	Details
200	<div><div>Response body</div><div><pre>{ "id": "3b5f286a-0e43-4ce9-ae95-5f651cd74f5b", "nome": "Fear of the Dark", "duracao": 719 }</pre></div><div>Download</div></div> <div><div>Response headers</div><div><pre>content-type: application/json; charset=utf-8 date: Sat,09 Dec 2023 18:41:09 GMT server: Kestrel</pre></div></div>

Responses

Code	Description	Links
200	Success	No links

Plano

GET

/api/Plano/{id}

Parameters

Cancel

Name	Description
id * required string(\$uuid) (path)	<div>8d044595-d4a6-4e1a-9f09-dab92205c71c</div>

Execute

Clear

Responses

Curl

```
curl -X 'GET' \
'https://localhost:7244/api/Plano/8d044595-d4a6-4e1a-9f09-dab92205c71c' \
-H 'accept: */*'
```

Request URL

https://localhost:7244/api/Plano/8d044595-d4a6-4e1a-9f09-dab92205c71c

Server response

Code	Details
200	<div><div>Response body</div><div><pre>{ "id": "8d044595-d4a6-4e1a-9f09-dab92205c71c", "nome": "Plano Basico Musica", "descricao": "Plano Basico", "valor": 0.0 }</pre></div><div>Download</div></div> <div><div>Response headers</div><div><pre>content-type: application/json; charset=utf-8 date: Sat,09 Dec 2023 18:46:00 GMT server: Kestrel</pre></div></div>

Responses

Code	Description	Links
200	Success	No links

