

FIA/P GRADUAÇÃO

SISTEMAS DE INFORMAÇÃO

MICROSERVICE AND WEB ENGINEERING

PROF. Thiago Xavier

Planejamento

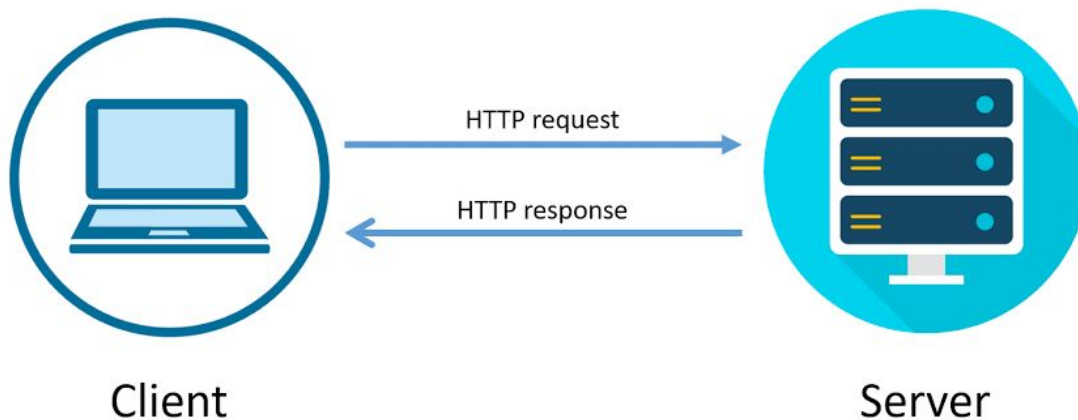
- Respostas status code;
- Validação

Respostas http

1. Quando o servidor recebe uma solicitação HTTP, o cliente deve saber o resultado da ação;
2. Temos vários códigos padronizados para aplicarmos em vários cenários;
3. Como boa prática podemos seguir alguns padrões para melhorarmos as comunicação dos serviços:
 - 2xx - Status de sucesso;
 - 3xx - Categoria de redirecionamento;
 - 4xx - Erro no cliente;
 - 5xx - Erro no server.

2xx - Sucesso

- 200 Ok - Padrão que representa sucesso. Resposta padrão para Get, quando há resultados;
- 201 Created - Indica que o recurso foi criado. Utilize para resposta do POST;
- 204 No Content - Indica que foi processado com sucesso, mas não há conteúdo para ser retornado. Pode ser utilizado no GET quando não há resultado, por exemplo;
- 202 Accepted - Indica que a requisição foi recebida, mas o processo será realizado de forma assíncrona;



3xx - Redirect

- Em geral é muito utilizado no frontend para redirecionamento para novo endereço;
- Em uma api restful não há grande utilidade.

4xx - Client error

- 400 Bad request: A solicitação não foi processada, pois o servidor não entendeu ou não enviou tudo o que é necessário para o processamento;
- 401 Unauthorized - Indica que o cliente não está autenticado e não pode acionar o recurso;
- 403 Forbidden - Indica que o cliente está autenticado, mas não tem permissão para acessar um recurso;
- 404 Not Found - indica que o recurso não foi localizado.

5xx - Erro no servidor

- 503 - O erro mais temido, indica que o servidor está fora do ar;
- 500 - Ocorreu alguma falha não tratada e que a requisição não foi processada.

Mão na massa

- Vamos ver alguns exemplo em node js;

- [https://quizizz.com/admin/quiz/64ef728dc4e96313ebc0af04?source=quiz_share;](https://quizizz.com/admin/quiz/64ef728dc4e96313ebc0af04?source=quiz_share)

Exercício

- Criar o serviços e seus respectivos endpoints da arquitetura já modela por nós. A ideia é que cada microserviço possa ter seu contexto delimitado, ou seja, pode ser que tenhamos mais de uma API.

Pré requisito:

- - Enviar todos os serviços para o git;
- - Desenvolvido em até 3 pessoas;
- - Definiremos os entregáveis no azure devops.

Serviços internos



Departamento

- código;
- nome;
- setor.

Pessoas

- Nome;
- documento;
- endereço;
- telefone.

Departamento pessoal

Folha salarial

- Código pessoa;
- salario;
- data;
- código
- departamento

Atendimento ao cliente

Tickets

- identificado atendente;
- título;
- telefone;
- identificador cliente

Atendentes

- identificado pessoa;
- identificado do setor;

Registro de atendimentos

- texto;
- ticket;