

EXERCÍCIOS DE REVISÃO

1. Uma característica importante de uma API Rest refere-se a um método de comunicação no qual o servidor completa cada solicitação do cliente independentemente de todas as solicitações anteriores, e os clientes podem solicitar recursos em qualquer ordem.

Essa característica é conhecida como:

- A) Ausência de estado.
- B) Capacidade de armazenagem.
- C) Independência.
- D) Interface uniforme.
- E) Sistema em camadas.

2. Na abordagem do Spring para construir serviços web RESTful, as solicitações HTTP são tratadas por um controlador, que é uma classe identificada com a anotação:

- A) @RequestController
- B) @RestController
- C) @RequestMapping
- D) @RestController
- E) @RestController

3. Consiste em uma arquitetura para sistemas, com base em redes (sistemas distribuídos), que é conceituada como um conjunto de constraints (restrições ou princípios) com o objetivo de definir a melhor forma de particionar um sistema. Essa arquitetura denomina-se:

- A) Streaming API
- B) REST.
- C) SPRING.
- D) GET.
- E) POST.

4. REST e SOAP são duas abordagens diferentes de transmissão de dados online. Especificamente, eles definem como as interfaces de programação de aplicações (APIs) são criadas, o que possibilita a comunicação dos dados entre aplicações web. Assinale a alternativa correta em relação ao assunto.

- A) Todas as mensagens SOAP são codificadas utilizando a formatação CSV.
- B) Quando uma solicitação de dados é enviada a uma API REST, ela sempre é realizada por meio do protocolo de transferência de arquivos FTP (File Transfer Protocol).
- C) SOAP é um protocolo leve para troca de informações em um ambiente centralizado utilizando o protocolo de transferência de arquivos FTP (File Transfer Protocol).

D) O SOAP pode ser usado em combinação com vários outros protocolos; no entanto, as únicas ligações definidas no protocolo descrevem como usar SOAP em combinação com os protocolos HTTP e TCP.

E) SOAP é um protocolo baseado em XML para troca de informações em um ambiente distribuído. É utilizado para troca de mensagens entre aplicativos distribuídos pela rede. Estes aplicativos, ou “Web services”, possuem uma interface de acesso simples e bem definida.

5. O REST é uma nova forma de criar web services que permite a interoperabilidade entre os sistemas. Sobre o REST é correto afirmar que:

- A) Restringe a comunicação entre aplicações.
- B) Permite a interoperabilidade entre sistemas.
- C) Impede o recebimento de mensagem de spam.
- D) Assegura a transferência de um pacote via FTP.
- E) Acelera a transferência de dados com o protocolo SMB.

6. Qual verbo HTTP é normalmente utilizado para criar um recurso em uma API?

- A) SUBMIT
- B) WRITE
- C) POST
- D) CREATE
- E) PUT

7. Sobre os códigos HTTP 200 e 201, eles representam qual tipo de retorno?

- A) 200 – sucesso genérico 201-sucesso na criação de um registro
- B) 200 - sucesso na criação de um registro; 201 - sucesso genérico
- C) 200 - sucesso generico; 201 - sucesso generico para metodos PUT
- D) 201 - erro na criação de registro; 200 – sucesso

8. Qual é o código HTTP retornado em caso de erro interno no servidor (Internal Server Error)?

- A) 500
- B) 501
- C) 504
- D) 502

9. Assinale a resposta CORRETA, sobre APIs síncronas:

- A) Seu fluxo de execução e retorno são independentes
- B) Trabalham em cadeia, chamando outras APIs para a mesma execução
- C) Executam uma chamada por execução
- D) São APIs consideradas ágeis

10. O técnico em informática Marcos implementou o web service REST **obtemPong** utilizando Java com framework Spring. O web service **obtemPong** recebe o parâmetro obrigatório **ping**.

Observe abaixo o principal trecho do código-fonte de obtemPong:

```
@GetMapping("/api/v1/pong") @ResponseBody
public String obtemPong(@RequestParam String ping) {
    return ping;
}
```

Para que o parâmetro **ping** deixe de ser obrigatório e automaticamente assuma o valor “**pong**” caso esteja ausente da mensagem de requisição, Marcos deve adicionar à anotação **@RequestParam** do parâmetro ping o argumento:

- A) value = “pong”;
- B) name = “pong”;
- C) defaultValue = “pong”;
- D) required = “ping?ping:pong”;
- E) value = “ping?ping:pong”.

11. O mecanismo de configuração automática baseado em convenção do Spring Boot pode ser iniciado anotando a classe que contém o método principal estático, com a anotação

- A) @SpringMainClass
- B) @SpringBootApplication
- C) @SpringMainClass
- D) @SpringBootApplication
- E) @SpringBootApplication

12. Considere a classe abaixo, criada utilizando Spring Boot, em condições ideais.

```
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.EnableAutoConfiguration;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;
```

```
@RestController
@EnableAutoConfiguration
public class Uninassau {
    @RequestMapping("/")
    String home() {
        return "Revisao";
    }
    public static void main(String[] args) {
        SpringApplication.run(LACUNA I, args);
    }
}
```

Para dizer à classe `SpringApplication` do Spring Boot qual é o componente primário do Spring, a **LACUNA** deve ser preenchida corretamente com:

- A) `SpringBoot.class`
- B) `Uninassau.class`
- C) `SpringBoot.start[Uninassau.class]`
- D) `Uninassau.app`
- E) `SpringBoot.main`