


Bacharelado em Ciência da Computação		<div style="text-align: right;">6,5</div> 
Disciplina: Gestão do Ciclo de Vida da Aplicação	Período: 4º	
Valor: 10 pontos (peso 0.4)	Data: 08/11/2024	
Professor: Rodrigo Martins Paoliães		
Discente: [REDACTED]		

Gabarito: Agradeço se você puder repassar suas respostas para quadro abaixo. Isso facilita durante minha correção.

1	2	3	4	5	6	7	8	9	10
D	B	D	C	D	D	D	E	A	A
11	12	13	14	15	16	17	18	19	20
D	C	D	B	A	A	D	D	C	A

1. Todas as afirmativas abaixo sobre o papel Product Owner no SCRUM são verdadeiras, EXCETO:

- a. O *Product Owner* é responsável por gerir o *Product Backlog* e assegurar o valor do trabalho feito pelo *Time Scrum*. ✗
- b. Este papel mantém o *Product Backlog* e assegura que ele esteja visível para todos envolvidos.
- c. O *Product Owner* pode ser um funcionário do cliente liberado pela direção a trabalhar em conjunto com o *Scrum Master* e *Time Scrum*. Quando o cliente não puder disponibilizar um funcionário próprio para atuar como *Product Owner*, a empresa responsável pelo desenvolvimento do *software* indica um *Product Owner* do seu quadro de pessoal que passa a estar em constante colaboração com o cliente, *Scrum Master* e *Time Scrum*.
- Ⓐ Gerente de produtos, representante do setor de marketing, inclusive CEOs podem exercer o papel de *Product Owner*.
- e. O *Product Owner* deve saber projetar soluções de que atendam aos requisitos do *software*

2. Assinale a alternativa INCORRETA sobre o Scrum Master

- a. O *Scrum Master* é o responsável por garantir que o *Time Scrum* se oriente pelos valores e práticas do *Scrum*.
- Ⓑ O *Scrum Master* realiza o papel de gerente de projetos usando de práticas tradicionais de gestão de projetos para liderar e dividir a carga de trabalho entre os membros da equipe, protegendo-os quando necessário.
- c. O *Scrum Master* protege o *Time Scrum* certificando-se de que os membros não se comprometam com compromissos além dos que eles conseguem cumprir dentro de uma *sprint*. ✓
- d. O *Scrum Master* facilita o *Daily Scrum* e se torna o responsável pela remoção de quaisquer obstáculos observados pelo *Time Scrum* durante estas reuniões. ✓
- e. O *Scrum Master* deve estar em constante comunicação com o *Time de Scrum* e com o *Product Owner*. ✓

3. Assinale a alternativa INCORRETA sobre o *Time Scrum (Scrum Team)*

- a. O *Time Scrum* é multifuncional: ele contém todas as especialidades necessárias para entregar o produto potencialmente implantável a cada *Sprint* e é auto organizável, com um alto grau de autonomia e responsabilidade.
- b. Um *Time Scrum* possui de 5 a 9 pessoas, mas é possível escalar utilizando um "*Scrum de Scrums*".

c. O *Time Scrum* tem de chegar a um acordo na sua definição de "Pronto" e compartilhá-las com o *Product Owner*.

☒ d. O *Time Scrum* estima o tamanho de cada item do *Product Backlog* bem como os prioriza para serem encaixados dentro de um *sprint*.

e. O *Time Scrum* quebra os itens do *Product Backlog* em tarefas estimadas em horas e que torne possível um compartilhamento de atividades mais efetivo.

4. Todas as afirmativas abaixo sobre o *Scrum* são verdadeiras, EXCETO:

a. O *Scrum* usa o conceito de *velocidade* para obter prazos a partir de uma lista de funcionalidades.

b. É importante ressaltar que a *velocidade* de um *Time Scrum* é otimizada com equipes consistentes de baixa rotatividade.

☒ c. Recomenda-se que o esforço estimado para realização dos itens do *Product Backlog* não esteja atrelado a uma unidade de tempo (esforço relativo).

d. No conceito de *velocidade* estão embutidos as restrições do ambiente, os riscos e a produtividade.

e. *Velocidade* é a quantidade de horas que o *Time Scrum* acredita conseguir trabalhar para implementar as *Histórias de Usuário* previstas para um *Sprint*.

5. Assinale a alternativa INCORRETA.

a. *Histórias de granularidade grossa* são chamadas de *epics*.

b. Temas são conjuntos de *epics*.

c. Uma história de usuário é uma história curta na perspectiva do usuário sobre o que ele acha valioso no produto.

☒ d. *Histórias de usuário* são substitutos de uma especificação de requisitos.

e. Uma história de usuário representa um trabalho de valor detalhado que Time de Desenvolvimento consegue entregar rapidamente.

6. Qual pergunta uma *user story* NÃO responde?

a. Quem

b. O quê

c. Por quê

☒ d. Como

e. Nenhuma das demais alternativas.

7. Assinale a alternativa incorreta sobre JDBC:

a. Uma URL JDBC inicia-se com a *substring* `jdbc:` X

b. A *interface* `Statement` é uma maneira eficiente de se utilizar consultas parametrizadas com JDBC.

c. `Statement` encapsula uma instrução SQL que é passada ao banco de dados para ser analisada, compilada, planejada e executada.

☒ d. O método `createStatement()` da interface `Connection` é usado para criar uma instrução.

e. O método `executeQuery()` da interface `Statement` é usado para executar consultas ao banco de dados. X

Assinale a alternativa INCORRETA sobre JDBC:

- a. JDBC permite a comunicação entre aplicativos Java e bancos de dados relacionais. ✗
- b. Para estabelecer uma conexão JDBC, é necessário especificar uma URL de banco de dados, nome de usuário e senha. ✗
- c. O `Statement` é usado para executar consultas SQL e não é adequado para consultas parametrizadas, pois está sujeito a ataques de injeção de SQL. ✗
- d. O método `executeQuery` é utilizado para executar consultas SQL que retornam dados, enquanto `executeUpdate` é usado para consultas que modificam dados.
- e. O JDBC fornece suporte nativo para manipulação de transações distribuídas (ou seja, transações que envolvem múltiplos bancos de dados ou recursos), sem necessidade de *frameworks* adicionais.

9. Comando para iniciar um repositório Git no diretório de trabalho:

- a. `git init`
- b. `git initialize`
- c. `git start`
- d. `git begin`
- e. `git init-repo`

10. O que faz o comando `git mv teste.txt teste2.txt`?

- a. Move os arquivos `teste.txt` e `teste2.txt` para a Stage Area (index).
- b. Move os arquivos `teste.txt` e `teste2.txt` para repositório.
- c. Renomeia o arquivo `teste.txt` para `teste2.txt`.
- d. Faz uma cópia do arquivo `teste.txt` com o nome de `teste2.txt`.
- e. Nenhuma das demais alternativas.

11. Qual o efeito do comando `git checkout -- teste.txt`?

- a. Sobrescreve o arquivo `teste.txt` no diretório de trabalho com o arquivo `teste.txt` vindo do repositório.
- b. Sobrescreve o arquivo `teste.txt` no diretório de trabalho com o arquivo `teste.txt` vindo da Stage Area (index).
- c. Compara o arquivo `teste.txt` no diretório de trabalho com o arquivo `teste.txt` vindo do repositório alertando o usuário existam diferenças entre eles.
- d. Compara o arquivo `teste.txt` no diretório de trabalho com o arquivo `teste.txt` vindo da Stage Area (index) alertando o usuário existam diferenças entre eles.
- e. Nenhuma das demais alternativas.

12. Complete a lacuna abaixo para fazer o *checkout* de um *commit* em particular do repositório.

`git checkout _____`

- a. Número inteiro do *commit* que se deseja recuperar.
- b. Código *hashcode* do *commit* que se deseja recuperar.
- c. Código SHA-1 do *commit* que se deseja recuperar. Embora seja um número de 40 caracteres, apenas os iniciais para identificação única do *commit* são suficientes.
- d. Chave RSA do *commit*.
- e. Nenhuma das demais alternativas.

13. Assinale a alternativa INCORRETA sobre JPA:

- a. Se estiver trabalhando com um banco de dados MySQL, recomenda-se usar `GenerationType.IDENTITY` para geração de chaves primárias numéricas de auto-incremento.
- b. Com JPA podemos alterar o mapeamento padrão para nomes de tabelas e colunas em um banco de dados, customizando os nomes de tabelas e colunas geradas.
- c. Antes do Java SE 8, usávamos as anotações `@Temporal` e suas constantes `TemporalType.TIMESTAMP` e `TemporalType.DATE` para mapearmos datas e horas para um banco de dados.
- ☒ d. Em uma aplicação Java SE, o JPA gerencia automaticamente as transações para cada operação de banco de dados, sem necessidade de configuração adicional.
- e. Em JPA, existem diversas estratégias para geração de chaves primárias em uma tabela de banco de dados.

14. Assinale a alternativa INCORRETA sobre JPA

- a. Podemos usar a anotação `@Table` para criarmos um nome personalizado para uma tabela no banco de dados.
- ☒ b. Tipos enumerados (*Enum*) não podem ser mapeados diretamente com JPA. Precisamos transformá-los em classes Java comum antes do mapeamento.
- c. Em ambientes de produção, diferentemente de ambientes de desenvolvimento, não é recomendável configurar o arquivo `persistence.xml` para remover as tabelas do banco de dados toda vez que você executa uma aplicação com JPA.
- d. JPA (Jakarta Persistence API) é um *framework* objeto-relacional para Java.
- e. O principal arquivo de configuração de um projeto JPA é conhecido como `persistence.xml`.

15. Qual das seguintes é uma anotação usada para mapear endpoints em um controlador REST no Spring Boot?

- ☒ a. `@Service`
- b. `@Entity`
- c. `@Repository`
- d. `@GetMapping`
- e. `@Component`

16. Qual anotação é utilizada para vincular os dados enviados de um formulário a um objeto em um controlador Spring?

- ☒ a. `@RequestParam`
- b. `@ModelAttribute`
- c. `@RequestMapping`
- d. `@PathVariable`
- e. `@RestController`

Assinale a alternativa **INCORRETA** sobre a criação de um projeto Spring Boot:

- a. O Spring Boot facilita o desenvolvimento de aplicativos Spring sem exigir muita configuração manual.
- b. A ferramenta web `start.spring.io` é utilizada para criar um projeto Spring Boot com as dependências desejadas.
- c. O `@RestController` define um ponto de acesso que responde a solicitações HTTP.
- ☒ d. A anotação `@GetMapping` serve para mapear requisições **POST** ao método sendo anotado.
- e. O projeto Spring Boot pode ser executado usando o comando `mvn spring-boot:run`

18. Assinale a alternativa **CORRETA** sobre o método `hello()` no exemplo de Spring Boot abaixo:

```
@RestController
public class HelloController {

    @GetMapping("/hello")
    public String hello(@RequestParam(defaultValue = "World") String name) {
        return String.format("Hello, %s!", name);
    }
}
```

- a. O método `hello()` responde apenas a solicitações com parâmetros obrigatórios. ✗
- b. O parâmetro `name` não tem um valor padrão definido no método `hello()`. ✗
- c. O `@RequestParam` define que o parâmetro `name` deve sempre ser fornecido pelo usuário. ✗
- ☒ d. O método `hello()` utiliza `String.format` para formatar a mensagem de resposta. ✓
- e. Nenhuma das demais alternativas.

19. Assinale a alternativa **INCORRETA** Sobre o uso de `JdbcTemplate` no desenvolvimento de aplicativos com Spring:

- a. `JdbcTemplate` simplifica a execução de comandos SQL como `SELECT`, `INSERT` e `UPDATE`.
- b. `JdbcTemplate` é compatível com bancos de dados relacionais e suporta consultas parametrizadas.
- ☒ c. A classe `JdbcTemplate` exige configuração manual de cada conexão com o banco de dados.
- d. `JdbcTemplate` pode ser usada junto com um banco de dados H2 em memória para testes rápidos.
- e. `JdbcTemplate` auxilia na redução de código repetitivo ao lidar com exceções SQL.

20. Assinale a alternativa **CORRETA** considerando o exemplo de uso de banco de dados relacional com Spring Boot:

- ☒ a. O arquivo `application.properties` pode ser usado para definir detalhes de conexão, como URL e credenciais do banco de dados.
- b. A anotação `@Entity` é usada para criar tabelas em um banco de dados H2 de forma automática.
- c. O H2 permite que dados sejam persistidos permanentemente por padrão.
- d. `JdbcTemplate` é a única forma de interagir com bancos de dados usando Spring. ✗
- e. Spring não permite o uso de bancos de dados em memória em ambientes de teste. >