

# Conhecimento Técnico

## Questões Técnicas:

### Questão 1

Implemente o método *isOn()* abaixo considerando as seguinte informações:

- O método *isOn()* deve chamar o método *static isOff(): boolean* que verifica se um determinado processo esta desligado.
- O método *isOn()* deve obrigatoriamente:
  - Se o processo estiver ligado, retornar true;
  - Senão, retornar false;

Implementação:

```
private static boolean statusProcesso = false;

private static boolean isOff() {
    return !statusProcesso;
}

private static boolean isOn() {
    return !isOff();
}
```

---

### Questão 2

Considere o seguinte cenário: o usuário do sistema está reclamando que quando um item gera erro no processamento, os demais que estão na lista e sem erro não são processados. Sendo assim, ajuste o método *processarItens* para processar todos itens mesmo quando existirem itens com erro na lista e gerar algum tipo de aviso técnico simples alertando para os itens com erro.

```
private void processarItens(List<Item> itens) {
try {
    for (Item i : itens) {
processar(i);
    }
    } catch (ItemException e) {
    }
}

private void processar(Item i) throws
ItemException { //implementação do processamento
omitido
}
```

```
    ...  
}
```

**Método ajustado:**

```
private void processarItens(List<Item> itens) {  
    for (Item i : itens) {  
        try {  
            processar(i);  
        } catch (ItemException e) {  
            //Em aplicações reais, deve ser gerado um log (Log4j)  
            System.out.println("Erro ao processar o item: " + i);  
        }  
    }  
}
```

---

### Questão 3

**Considere que você precisará efetuar uma correção de código em um sistema para resolver um chamado/ticket e gerar uma versão para implantação em Produção. Seguindo as dicas abaixo, cite os passos (fluxo) que julga correto do ponto de vista de gestão de configuração e gestão de versão de código:**

Dicas

- O repositório utilizado é SVN.
- O repositório do sistema é subdividido em: TRUNK, BRANCHES e TAGS.
- Na equipe há outros desenvolvedores atuando em outras correções no mesmo sistema.
- Descreva quantos passos achar necessário até a implantação em produção.
- Não é necessário descrever como executará cada passo, apenas quais passos serão executados.

Passo 1 – Verificar em qual repositório está a versão em Produção onde deve haver a correção.

Passo 2 – Criar uma nova branch a partir da versão do passo 1 (master\_correcao\_123456)

Passo 3 – Realizar a correção na branch master\_correcao\_123456

Passo 4 – Comitar e submeter a teste a versão do passo anterior

Passo 5 – Caso aprovado, gerar versão a partir da branch master\_correcao\_123456

Passo 6 – Gerar TAG da versão do Passo 1

Passo 7 – Realizar o merge da master com a master\_correcao\_123456

Passo 8 – Comitar branch do Passo 1

---

**Questão 4**

Considerando as tabelas abaixo, faça uma consulta SQL (banco Oracle) que retorne o Nome e Salário do CPF 123.123.123-12 em 01/06/2018.

SALARIO				
ID_SALARIO (PK)	ID_PESSOA (FK)	SALARIO	DATA_INICIO_VIGENCIA	DATA_FIM_VIGENCIA
1	1	1000,00	01/05/2017	01/01/2018
2	1	2000,00	01/01/2018	01/11/2018
3	1	2150,00	01/11/2018	NULL
4	2	1500,00	01/02/2016	NULL

PESSOA		
ID_PESSOA (PK)	CPF	NOME
1	123.123.123-12	João da Silva
2	456.456.456-01	José da Silva Júnior

**Query:**

```
SELECT NOME, s.SALARIO FROM SALARIO sal
INNER JOIN PESSOA pes ON pes.ID_PESSOA = sal.ID_PESSOA
WHERE pes.CPF = '123.123.123-12'
AND sal.DATA_INICIO_VIGENCIA <= TO_DATE('01-JUN-2018','DD-MON-YYYY')
AND sal.DATA_FIM_VIGENCIA >= TO_DATE('01-JUN-2018','DD-MON-YYYY')
```

---

### Questão 5

Para as tabelas abaixo, faça uma consulta SQL (Oracle) que retorne:

a) o total de remuneração do funcionário (salário + todos benefícios);

BENEFICIO	
CD_BENEFICIO (PK)	Nu
CD_PESSOA (FK)	Nu
VALOR_BENEFICIO	Nu

PESSOA	
CD_PESSOA (PK)	Numérico
NOME	Alfanumérico
SALARIO	Numérico

#### Query:

```
SELECT pes.NOME, (SUM(ben.VALOR_BENEFICIO) + pes.SALARIO) as 'Total  
Remuneração'
```

```
FROM PESSOA pes
```

```
INNER JOIN BENEFICIO ben on ben.CD_PESSOA = pes.CD_PESSOA
```

```
GROUP BY CD_PESSOA;
```

---

### **Questão 6**

**O que é e para que é usado a Alternate Key em uma tabela de banco de dados?**

Usado para índices. Alternate Key são chaves candidatas para “substituir” a chave primaria em um banco. É como se fosse utilizada para diferenciar registros dentro de um conjunto de dados.

### **Questão 7**

**Explique a principal diferença entre AngularJS e Angular?**

Angular JS é o Angular 1, primeira versão do framework e que foi descontinuado em 2016. As versões posteriores do AngularJS (ou Angular 1) são chamadas somente de Angular e possuem diferenças significativas de codificação e uso de suas bibliotecas.

---

### **Questão 8**

**Descreva as principais utilidades do Maven no processo de build de código.**

O Maven define quais e bibliotecas são utilizadas pelo projeto e qual a versão das bibliotecas utilizadas, funcionando como gerenciador de repositórios e podendo ir além disso.

Para o desenvolvimento, ele baixa as dependências utilizadas e configuradas no arquivo pom.xml e pode realizar a compilação e geração da release.

---

### **Questão 9**

**Descreva quais são os métodos HTTP e para que são utilizados no protocolo REST.**

GET -> Consulta de dados (somente leitura)

PUT -> Criar ou atualizar uma informação identificada por uma URL (escrita)

DELETE -> Ação contrária do PUT. Utilizada quando se deseja deletar o recurso

POST -> Ação similar ao POST. Pode resultar em modificação de recurso (escrita)

---

### Questão 10

Descreva seu conhecimento em Spring Batch para os elementos abaixo:

Obs: Procure descrever o que são e para que são utilizados na arquitetura Spring Batch.

1. **Job:** Job é uma aplicação que processa uma quantidade finita de dados sem interação ou interrupção.
2. **Step:** São os passos que serão executados em uma Job.

---

### Questão 11

Considerando seu conhecimento técnico, favor informar qual nível de conhecimento você possui em cada assunto abaixo:

Régua

1. Desconheço totalmente
2. Já ouvi falar
3. Li artigos sobre o assunto, mas não me lembro do que se trata
4. Conheço o assunto e posso explicar resumidamente do que se trata
5. Conheço o assunto e usei em uma ocasião
6. Usei em várias ocasiões há muito tempo e atualmente preciso de apoio no assunto
7. Usei em várias ocasiões recentemente, mas ainda preciso de apoio no assunto
8. Me sinto seguro no assunto para usar quando necessário com pouco ou nenhum apoio
9. Domino o assunto e me sinto seguro usar quando necessário com total independência
10. Domino e posso treinar o time no assunto

Assunto	Nível	Assunto	Nível	Assunto	Nível
Futebol (exemplo)	8	Maven	9	Spring MVC	10
Curling (exemplo)	2	SVN	10	Spring Batch	8
Javascript / JQuery	8	Git	10	Spring Data	8
Ajax / REST	10	WebSphere	8	Jenkins	7
Angular 2	8	SpringBoot	8	Enterprise Architect	3