**Nome: Vinicius Thomé Vallado Data: 20/04/2022**

**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:

o Se o processo estiver ligado, retornar true;

o Senão, retornar false;

Implementação:

**public** **class** Teste {

**public** **static** **void** main(String[] args) {

**if** (*isOn*()) {

System.***out***.println("Processo em execução...");

}

}

**public** **static** **boolean** isOn() {

**return** (*isOff*()) ? **false** : **true**;

}

**public** **static** **boolean** isOff() {

**if** (*processo*() > 5) {

**return** **false**;

}

**return** **true**;

}

**public** **static** **int** processo() {

**int** min = 0;

**int** max = 10;

**return** (**int**)Math.*floor*(Math.*random*()\*(max-min+1)+min);

}

}

**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) {

t**ry** {

*processar*(i);

} **catch** (ItemException e) {

System.***out***.println(e.getMessage());

}

}

}

**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 – Criar um repositório local;

Passo 2 – Criar uma BRANCH a partir da URL onde estão os arquivos desejáveis (TRUNK);

Passo 3 – Realizar um checkout e escolher o caminho que a branche foi criada.

Passo 4 – Ao terminar as alterações, realizar um commit.

Passo 5 – Para realizar o merge a partir da pasta TRUNK

**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.

Query:

-- Consulta SQL (banco ORACLE) que retorne o Nome e Salário do CPF 123.123.123-12 em 01/06/2018.

**SELECT** P.NOME, S.SALARIO, P.CPF, DATA\_INICIO\_VIGENCIA

**FROM** PESSOA P

**INNER** **JOIN** SALARIO S

**ON** P.ID\_PESSOA = S.ID\_PESSOA

**WHERE** P.CPF = '123.123.123-12'

**AND** DATA\_INICIO\_VIGENCIA = TO\_DATE('01/06/2018','DD/MM/YYYY');

-- Consulta SQL (banco MySQL) que retorne o Nome e Salário do CPF 123.123.123-12 em 01/06/2018.

**SELECT** P.NOME, S.SALARIO, P.CPF, DATA\_INICIO\_VIGENCIA

**FROM** `teste-app`.PESSOA P

**INNER** **JOIN** `teste-app`.SALARIO S

**ON** P.ID\_PESSOA = S.ID\_PESSOA

**WHERE** P.CPF = '123.123.123-12'

**AND** DATA\_INICIO\_VIGENCIA = STR\_TO\_DATE('01/06/2018','%d/%m/%Y');

-- Consulta SQL (banco Oracle) que retorne o Nome e Salário do CPF 123.123.123-12 em 01/05/2017.

**SELECT** P.NOME, S.SALARIO, P.CPF, DATA\_INICIO\_VIGENCIA,STR\_TO\_DATE('01/05/2017','%d/%m/%Y')

**FROM** PESSOA P

**INNER** **JOIN** SALARIO S

**ON** P.ID\_PESSOA = S.ID\_PESSOA

**WHERE** P.CPF = '123.123.123-12'

**AND** DATA\_INICIO\_VIGENCIA = TO\_DATE('01/05/2017','DD/MM/YYYY');

-- Consulta SQL (banco MySQL) que retorne o Nome e Salário do CPF 123.123.123-12 em 01/05/2017.

**SELECT** P.NOME, S.SALARIO, P.CPF, DATA\_INICIO\_VIGENCIA,STR\_TO\_DATE('01/05/2017','%d/%m/%Y')

**FROM** `teste-app`.PESSOA P

**INNER** **JOIN** `teste-app`.SALARIO S

**ON** P.ID\_PESSOA = S.ID\_PESSOA

**WHERE** P.CPF = '123.123.123-12'

**AND** DATA\_INICIO\_VIGENCIA = STR\_TO\_DATE('01/05/2017','%d/%m/%Y');

**Questão 5**

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

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

Query:

-- Consulta SQL (banco Oracle)

**SELECT** P.ID\_PESSOA, P.NOME, S.SALARIO, TB.TOTAL\_BENEFICIOS, TB.TOTAL\_BENEFICIOS + S.SALARIO

**FROM** PESSOA P

**INNER** **JOIN** SALARIO S

**ON** P.ID\_PESSOA = S.ID\_PESSOA

**INNER** **JOIN**

(**SELECT** PE.ID\_PESSOA, SUM(BE.VALOR\_BENEFICIO) TOTAL\_BENEFICIOS

**FROM** PESSOA PE

**INNER** **JOIN** SALARIO SA

**ON** PE.ID\_PESSOA = SA.ID\_PESSOA

**INNER** **JOIN** BENEFICIO BE

**ON** PE.CD\_PESSOA = CD.ID\_PESSOA

**WHERE** SA.DATA\_FIM\_VIGENCIA **IS** **NULL**

**GROUP** **BY** PE.ID\_PESSOA, PE.NOME) TB

**ON** P.ID\_PESSOA = TB.ID\_PESSOA

**WHERE** S.DATA\_FIM\_VIGENCIA **IS** **NULL**;

-- Consulta SQL (banco MySQL)

**SELECT** P.ID\_PESSOA, P.NOME, S.SALARIO, TB.TOTAL\_BENEFICIOS, TB.TOTAL\_BENEFICIOS + S.SALARIO

**FROM** `teste-app`.PESSOA P

**INNER** **JOIN** `teste-app`.SALARIO S

**ON** P.ID\_PESSOA = S.ID\_PESSOA

**INNER** **JOIN**

(**SELECT** PE.ID\_PESSOA, SUM(BE.VALOR\_BENEFICIO) TOTAL\_BENEFICIOS

**FROM** `teste-app`.PESSOA PE

**INNER** **JOIN** `teste-app`.SALARIO SA

**ON** PE.ID\_PESSOA = SA.ID\_PESSOA

**INNER** **JOIN** `teste-app`.BENEFICIO BE

**ON** PE.ID\_PESSOA = BE.ID\_PESSOA

**WHERE** SA.DATA\_FIM\_VIGENCIA **IS** **NULL**

**GROUP** **BY** PE.ID\_PESSOA, PE.NOME) TB

**ON** P.ID\_PESSOA = TB.ID\_PESSOA

**WHERE** S.DATA\_FIM\_VIGENCIA **IS** **NULL**;

**Questão 6**

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

*Alternate Key são chaves ou conjunto de atributos únicos . (ChavesCandidatas – modelo conceitual), identificam exclusivamente uma linha em uma determinada tabela, podendo haver várias Alternates Keys na mesma. Portanto uma delas pode vir a tornar uma chave primária no modelo logico do banco de dados.*

*São características fundamentais para ser uma Alternate Key:*

*- Uma chave primária não pode ser uma Altenate Key. Para uma tabela com uma única Alternate Key esta deve ser a Chave Primária e não haverá nenhuma Alternate Key na mesma.*

*- Alternate Keys não poder ser chaves estrangeiras.*

*- Uma Alternate Key deve ser exclusiva na tabela.*

*- Uma Alternate Key pode receber valor null.*

**Questão 7**

Explique a principal diferença entre AngularJS e Angular?

*O AngularJS é um front-end baseado em javascript de estrutura dinâmica usado para desenvolver aplicações de página única.*

*O Angular 2 é totalmente diferente do AngularJS. Os recursos como controladores, $scope, $scope variables foram substituídos pelos componentes e diretivas. Um componente encapsula uma estrutura HTML, estilo (CSS) e comportamento (Javascript) e podem ser utilizadas como TAGS HTML customizadas.*

*Pode-se escrever uma aplicação Angular utilizando TypeScripy ou Dart, sendo o primeiro mais utilizado. Sendo assim permitisse codificar com muito mais facilidade os conceitos de OOP. É possível utilizar lambda, generics, heranças, interfaces... Quando a aplicação Angular é compilada (“Transpile”) que converte o TypeScript para JavaScript, sendo possível debugar o código TypeScript.*

*O Angular possui um mecanismo de injeção de dependência nativo, possui o ferramental Angular CLI, WebPack, Reative Forms e pode ser usado em aplicações hibridas para Mobile.*

*Demais características de ambos frameworks:*

*AngularJS :*

*- Utiliza a arquitetura Model-View-Control por natureza, sendo assim os componentes são construídos separadamente e posteriormente combinados.*

*- Usa tags HTML básicas para criar a interface com o usuário.*

*- Ajuda na criação de um sistema de paginação e na filtragem da matriz de dados em relação aos parâmetros fornecidos.*

*- As expressões são escritas dentro de duas chaves (“{{expressão}}”), sendo estes úteis na vinculação de dados bidirecional. Sendo assim, ajuda a criar a sincronização entre o modelo e a exibição. As alteraçõe feita no modelo são refletidas instantaneamente na visualização e as alterações na visualização são refletidas nos modelos.*

*Angular 2:*

*- Útil no desenvolvimento de aplicativos móveis responsivos.*

*- O recurso de carregamento dinâmico ajuda a reduzir o tempo de carregamento.*

*- Possui serviço de localização e modelo de navegação que melhoram o serviço de roteamento.*

**Questão 8**

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

*O maven é responsável pelo gerenciamento de dependências incluindo atualização automática e fechamento de dependências (também conhecido como dependências transitivas). O Maven é baseado em compilações baseadas de modelo sendo capaz de compilar qualquer número de projetos em tipos de saída predefinidos, como JAR, WAR ou distribuição com base em metadados sobre o projeto. O Maven integra ao seu sistema de controle de origem (como Subversion ou Git) e gerenciará o lançamento de um projeto com base em uma determinada tag. Ele também pode publicar isso em um local de distribuição para uso por outros projetos. O Maven é capaz de publicar saídas individuais, como um JAR, um arquivo incluindo outras dependências e documentação, ou como uma distribuição de origem.*

**Questão 9**

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

1. *GET: Retorna a apresentação de um recurso. O método GET por definição jamais deve criar, atualizar ou excluir um recurso. O resultado do método será o mesmo para um determinado conjunto de informações, sendo assim o método é safe e idempotente.*
2. *POST: Cria um recurso. Toda vez que é chamado um recurso pode ser criado, por isso não é safe muito menos idempotente pois pode retornar parâmetros diferente na resposta.*
3. *PUT: Atualiza um recurso mas o seu retorno é sempre igual, independente de quantas vezes for chamado. Não é safe mas idempotente.*
4. *DELETE: Remove um recurso, mas o seu retorno é sempre igual, independente de quantas vezes for chamado. Não é safe mas idempotente.*
5. *PATCH: Atualiza um recurso mas diferente do PUT não é idempotente.*
6. *OPTIONS: Retorna os verbos HTTP de um recurso.*
7. *HEAD: Busca apenas um cabeçalho de um recurso.*

**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: É uma entidade que encapsula um processo de lote inteiro, um contâiner para as instâncias de Step. Combina várias etapas em um fluxo que permite configurações de propriedades globais para todas as etapas. Cada configuração de trabalho contém o nome simples do job, definição e ordenação das instâncias do step e se o trabalho é reinicíavel ou não.*

*Uma instancia de um job refere-se ao conceito de execução lógica. Apenas uma instancia corresponde a um determinado job.*

1. Step: É um objeto de domínio que encapsula uma fase sequencial independente de um trabalho em lote. Portanto, todo Job é composto inteiramente de uma ou mais steps (etapas) . Uma Etapa contém todas as informações necessárias para definir e controlar o processamento em lote real.

**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 |
| Javascript / JQuery | 9 | **Git** | 9 | **Spring Batch** | 5 |
| Ajax / REST | 9 | **WebSphere** | 6 | **Spring Data** | 9 |
| Angular 2 | 8 | **SpringBoot** | 9 | **Jenkins** | 8 |
| Maven | 9 | **Spring MVC** | 9 | **Enterprise Architect** | 6 |
| SVN | 7 |  | | | |