

[Painel](#) / [Meus cursos](#) / [CETEJ8.2020 01.JAVA XIX](#) / [25.janeiro - 31.janeiro](#) / [Questionário 2](#)

<b>Iniciado em</b>	sábado, 6 fev 2021, 13:48
<b>Estado</b>	Finalizada
<b>Concluída em</b>	sábado, 6 fev 2021, 13:59
<b>Tempo empregado</b>	11 minutos 2 segundos
<b>Notas</b>	1,0/1,0
<b>Avaliar</b>	9,5 de um máximo de 10,0(95%)

## Questão 1

Parcialmente correto

Atingiu 1,0 de 1,0

Nesta única questão você deverá preencher o formulário comparando as tecnologias [JaxP](#), JDOM e [JaxB](#). Cada linha da tabela contém o critério, as 3 tecnologias e também uma explicação sobre o critério empregado

ID - Critério	JDOM	SAX ( <a href="#">JAXB</a> )	<a href="#">JAXB</a>	Descrição d critério
1 - Nível da API	Média ✓	Baixa ✓	Alta ✓	Como o desenvolvedor acessa o documento XML Quanto mais baixa a API mais próximo programador está do código XML
2 - Capacidade de processamento (Vazão)	Alta ✗	Média ✓	Baixa ✓	Processar um volume massivo de XML. Processamento de E-books, por exemplo
3 - Desempenho (tempo de execução)	Média ✓	Média ✗	Alta ✓	Performance a processar um único documento XML com pouco volume de dados
4 - Capacidade de mapear XML em objetos Java (Marshalling / UnMarshalling)	Não ✓	Não ✓	Sim ✓	A API tem a capacidade de fazer a conversão de XML para Objetos e/ou Classes Java
5 - Implementação baseada em sequencia de eventos	Não ✓	Sim ✓	Não ✓	O uso da API depende da implementação de métodos Callback

ID - Critério	JDOM	SAX ( <a href="#">JAXB</a> )	<a href="#">JAXB</a>	Descrição d critério
6 - A API provê elementos XML com chaves Hash	<div>Sim</div> <div>✓</div>	<div>Não</div> <div>✓</div>	<div>Sim</div> <div>✓</div>	A API tem a capacidade de gerar e de associar um elemento XML chave Hash
7 - A abordagem é baseado em uma estrutura de árvore	<div>Sim</div> <div>✓</div>	<div>Não</div> <div>✓</div>	<div>Estrutura de Classes/Objetos</div> <div>✓</div>	Como a API aborda o XML. Como o documento XML é representado dentro do código Java
8 - Os dados são manipulados em memória	<div>Sim</div> <div>✓</div>	<div>Sim</div> <div>✓</div>	<div>Sim</div> <div>✓</div>	A modificação de algum elemento XML exige que ao menos tal elemento esteja em memória
9 - Utiliza Java annotations	<div>Não</div> <div>✓</div>	<div>Não</div> <div>✓</div>	<div>Sim</div> <div>✓</div>	A API faz uso de Annotations para amparar o desenvolvedor durante a codificação
10 - Utiliza o padrão Factory	<div>Sim</div> <div>✓</div>	<div>Sim</div> <div>✓</div>	<div>Sim</div> <div>✓</div>	A API precisa apenas da assinatura do método principal para criar o Parser
11- Utiliza o padrão Builder	<div>Sim</div> <div>✓</div>	<div>Não</div> <div>✓</div>	<div>Sim</div> <div>✓</div>	A API exige que sejam informados parâmetros na criação do Parser principal e com isso consegue controlar sua granularidade

ID - Critério	JDOM	SAX ( <a href="#">JAXB</a> )	<a href="#">JAXB</a>	Descrição do critério
16 - Acesso Aleatório	<input type="text" value="Sim"/> ✓	<input type="text" value="Não"/> ✓	<input type="text" value="Sim"/> ✓	A API pode acessar qualquer elemento XML de maneira aleatória. Se isso não for possível, então a API acessa o documento de maneira serial. Se o acesso aleatório é provido, então a navegação no XML é bidirecional.
17 - Análise parcial	<input type="text" value="Não"/> ✓	<input type="text" value="Sim"/> ✓	<input type="text" value="Sim"/> ✓	A API não depende do desenvolvedor para analisar parcialmente um documento.
18 - Volume de código gerado	<input type="text" value="Média"/> ✓	<input type="text" value="Alta"/> ✓	<input type="text" value="Baixa"/> ✓	Quanto de esforço e código um desenvolvedor precisa para programar o acesso de um mesmo XML.

Sua resposta está parcialmente correta.

Você selecionou corretamente 40.

A resposta correta é:

Nesta única questão você deverá preencher o formulário comparando as tecnologias [JaxP](#), JDOM e [JaxB](#). Cada linha da tabela contém o critério, as 3 tecnologias e também uma explicação sobre o critério empregado.

ID - Critério	JDOM	SAX ( <a href="#">JAXB</a> )	<a href="#">JAXB</a>	Descrição do critério
1 - Nível da API	[Média]	[Baixa]	[Alta]	Como o desenvolvedor acessa o documento XML. Quanto mais baixa a API, mais próximo o programador está do código XML.
2 - Capacidade de processamento (Vazão)	[Baixa]	[Média]	[Baixa]	Processar um volume massivo de XML. Processamento de E-books, por exemplo.
3 - Desempenho (tempo de execução)	[Média]	[Baixa]	[Alta]	Performance ao processar um único documento XML com pouco volume de dados.

ID - Critério	JDOM	SAX (JAXB)	JAXB	Descrição do critério
4 - Capacidade de mapear XML em objetos Java ( Marshalling / UnMarshalling)	[Não]	[Não]	[Sim]	A API tem a capacidade de fazer a conversão de XML para Objetos e/ou Classes Java
5 - Implementação baseada em sequencia de eventos	[Não]	[Sim]	[Não]	O uso da API depende da implementação de métodos Callback
6 - A API provê elementos XML com chaves Hash	[Sim]	[Não]	[Sim]	A API tem a capacidade de gerar e de associar um elemento XML à chave Hash
7 - A abordagem é baseado em uma estrutura de árvore	[Sim]	[Não]	[Estrutura de Classes/Objetos]	Como a API aborda o XML. Como o documento XML é representado dentro do código Java
8 - Os dados são manipulados em memória	[Sim]	[Sim]	[Sim]	A modificação de algum elemento XML exige que ao menos tal elemento esteja em memória
9 - Utiliza Java annotations	[Não]	[Não]	[Sim]	A API faz uso de Annotations para amparar o desenvolvedor durante a codificação
10 - Utiliza o padrão Factory	[Sim]	[Sim]	[Sim]	A API precisa apenas da assinatura do método principal para criar o Parser
11- Utiliza o padrão Builder	[Sim]	[Não]	[Sim]	A API exige que sejam informados parâmetros na criação do Parser principal e com isso consegue controlar sua granularidade
16 - Acesso Aleatório	[Sim]	[Não]	[Sim]	A API pode acessar qualquer elemento XML de maneira aleatória. Se isto não for possível, então a API acessa o documento de maneira serial. Se o acesso aleatório é provido então a navegação no XML é bidirecional
17 - Análise parcial	[Não]	[Sim]	[Sim]	A API não depende do desenvolvedor para analisar parcialmente um documento
18 - Volume de código gerado	[Média]	[Alta]	[Baixa]	Quanto de esforço e código um desenvolvedor precisa para programar o acesso de um mesmo XML

◀ Jaxb3XML

Seguir para...

Microserviços, por Martin Fowler e James Lewis – Pedro Mendes ►