

Certificado – Semana OmniStack 11.0

Total points 24/25 ?

Chegou o momento de mostrar que você devorou todo o conteúdo da Semana OmniStack 11.0 e está pronto para receber o seu certificado.

Mas vale lembrar que certificados não acrescentam nada na sua carreira. O que vale mesmo é o seu conhecimento, é dominar as tecnologias e ser capaz de construir suas próprias aplicações.

Então coloque o código da aplicação que você construiu durante essa semana lá no Github, compartilhe o que você aprendeu lá no LinkedIn, e principalmente, continue estudando sempre.

Você tem até domingo (29/03) às 23h59 para responder esse questionário.

E para ser aprovado e receber sua certificação, é necessário acertar pelo menos 70% das questões.

Boa avaliação!

Email address *

rogerio3@gmail.com

Seu Nome Completo *

ROGERIO DE SOUZA



✓ Quais desses são critérios para a escolha da stack? *

1/1

Selecione todas as alternativas corretas

- ☒ O que o mercado está procurando ✓
- ☒ O tamanho da comunidade e do ecossistema ao redor dessas tecnologias ✓
- ☐ A possível vida útil dessas ferramentas
- ☒ Quem está por trás do desenvolvimento delas ✓
- ☒ Quais são os casos de uso das tecnologias ✓
- ☒ Quantidade de vagas disponíveis no mercado ✓
- ☐ Se é uma tecnologia nova
- ☒ Quais empresas e quantas empresas estão utilizando ✓
- ☐ A tecnologia tem uma logomarca bonita

✗ Para você ter acesso às melhores vagas, é preciso: *

0/1

Selecione todas as alternativas corretas

- ☐ Conhecer o máximo de tecnologias diferentes
- ☒ Dominar um conjunto específico de tecnologias e ser capaz de resolver problemas com elas ✓
- ☒ Estar presente no LinkedIn e Github com conteúdos relevantes que mostrem sua evolução ✓
- ☒ Conhecer todos conceitos de como as tecnologias funcionam por baixo dos panos ✗

Correct answer

- ☒ Dominar um conjunto específico de tecnologias e ser capaz de resolver problemas com elas
- ☒ Estar presente no LinkedIn e Github com conteúdos relevantes que mostrem sua evolução



✓ Qual a melhor forma de acertar na escolha de uma stack? *

1/1

Selecione a alternativa correta

- ☐ Investir muito tempo experimentando cada tecnologia
- ☒ Se espelhar em alguém que já chegou onde você quer chegar e te ajude a fazer essa escolha ✓
- ☐ Fazer um curso em cada linguagem e ver qual você gostou mais

✓ O que é o Node.js? *

1/1

Selecione a alternativa correta

- ☐ Uma linguagem de programação
- ☒ Uma plataforma escrita em cima do motor V8 do Google Chrome ✓
- ☐ Uma biblioteca para o front-end
- ☐ Um framework para back-end

✓ Quais desses são de responsabilidade de um back-end? *

1/1

Selecione as alternativas corretas

- ☒ Regras de negócio ✓
- ☐ Estilização das páginas
- ☒ Conexão com Banco de Dados ✓
- ☐ Exibição da interface
- ☒ Comunicação com WebServices ✓
- ☐ Fazer café



✓ Qual é a estrutura de dados utilizada para conectar o back-end com outros clientes? * 1/1

Selecione a alternativa correta

☒ JSON



☐ JavaScript

☐ HTML

✓ Qual a abordagem do React? * 1/1

Selecione a alternativa correta

☐ A cada requisição o servidor retorna o conteúdo completo da página, em HTML e CSS

☒ Uma abordagem SPA, onde as requisições trazem apenas dados como resposta e, com esses dados, o front-end pode preencher as informações em tela. ✓

☐ Nenhuma das alternativas

✓ Qual a abordagem do React Native? * 1/1

Selecione a alternativa correta

☒ Todo código é feito em JavaScript e os próprios dispositivos passam a entender esse código JavaScript para gerar uma interface Nativa ✓

☐ Uma aplicação é desenvolvida para IOS e outra para Android

☐ O dispositivo cria uma webview (navegador embutido) que mostra exclusivamente HTML/CSS e JavaScript na página



✓ **O que é Expo? ***

1/1

Selecione a alternativa correta

- ☐ Expo é uma ferramenta que possibilita que interface nativa seja gerada apenas pelo JavaScript
- ☒ Expo é uma ferramenta que nos permite desenvolver para React Native sem instalar recursos nativos do Android e iOS ✓

✓ **Para que servem as migrations no back-end? ***

1/1

Selecione as alternativas corretas

- ☒ Para controle de versão do banco de dados ✓
- ☐ As migrations servem para conectarmos o back-end com o banco de dados
- ☒ Para facilitar o trabalho de mais devs no projeto, mantendo a base de dados sempre atualizada ✓
- ☐ As migrations servem apenas para criar tabelas no banco em ambiente de desenvolvimento

Sobre os tipos de parâmetro, marque a definição de cada um: *

	Parâmetros nomeados enviados na rota após o "?" usados em filtros e paginações, por exemplo.	Corpo da requisição utilizado para criar ou alterar recursos	Parâmetros utilizados para identificar recursos	Score	
Query Params	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	1/1	✓
Route Params	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	1/1	✓
Request Body	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	1/1	✓



Sobre os métodos HTTP, marque a definição de cada um deles: *

	Deletar uma informação	Alterar uma informação	Buscar ou listar uma informação	Criar uma informação	Score	
GET	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	1/1	✓
POST	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	1/1	✓
PUT	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	1/1	✓
DELETE	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	1/1	✓

✓ Como podemos definir as Entidades no back-end? *

1/1

Selecione a alternativa correta

- ☐ Dados que serão armazenados na aplicação mas que não são armazenados no banco de dados
- ☐ Arquivos que armazenam as regras de negócio da aplicação
- ☒ A representação de uma tabela do banco de dados, algo que será mantido pela aplicação ✓

✓ Sobre o conceito de Propriedades no React.js *

1/1

Selecione a alternativa correta

- ☐ São informações que o componente armazena e manipula
- ☒ São atributos repassados de um componente pai para o filho ✓
- ☐ São informações repassadas de um componente filho para o componente pai



✓ Sobre o conceito de Imutabilidade no React.js, selecione a alternativa correta * 1/1

Selecione a alternativa correta

- ☐ Segundo este conceito, podemos manipular o valor de um estado livremente
- ☐ Segundo este conceito, toda vez que um estado na aplicação sofre alteração, o componente sofre uma nova renderização
- ☒ Segundo este conceito, não podemos manipular o valor de um estado de uma forma direta. Podemos apenas sobrepor o valor da variável do estado ✓

✓ Sobre o conceito de Estado no ReactJS, selecione a alternativa correta * 1/1

Selecione a alternativa correta

- ☒ Trata-se de uma informação a ser mantida pelo componente. Toda vez que o estado é alterado, o componente faz a renderização novamente exibindo as novas informações em tela. ✓
- ☐ Trata-se de uma informação recebida de um componente pai
- ☐ O estado serve apenas para armazenarmos valores provindos de uma API

✓ Como o Expo funciona? * 1/1

Selecione a alternativa correta

- ☐ Precisamos instalar o Expo na nossa máquina e a única forma de visualizar é utilizando um Emulador como o Geny Motion
- ☒ Com o Expo, nós instalamos um aplicativo no celular chamado Expo, e dentro dele, tudo o que precisamos para desenvolver no React Native já está instalado. ✓
- ☐ De forma mágica e celestial, criando aplicações de forma totalmente automática



✓ Das opções abaixo, quais representam diferenças entre ReactJS e React Native? * 1/1

Selecione todas as alternativas corretas

- ☒ No ReactJS, os elementos em tela possuem um significado semântico. Já no React Native, não há semântica nos elementos ✓
- ☒ No React Native sempre precisamos passar uma tag style que recebe um objeto com a estilização. Já no ReactJS, podemos definir classes estilizadas ✓
- ☐ As propriedades no ReactJS são escritas em camel case (backgroundColor). No React Native, são escritas usando hifen. (background-color)
- ☐ No React não existe herança de estilos. No React Native, existe.

✓ O que é TDD? * 1/1

Selecione a alternativa correta

- ☐ Test-Driven Development - Segundo esta técnica fazemos os testes logo após o finalização do desenvolvimento das funcionalidades.
- ☒ Test-Driven Development - Segundo esta técnica fazemos os testes antes mesmo de terminar o desenvolvimento das funcionalidades. ✓
- ☐ Test-Driven Development - Segundo esta técnica fazemos os testes após a aplicação estar pronta, interagindo com a aplicação como um usuário final.
- ☐ Traduzindo a sigla de TDD temos: Desenvolvimento dirigido à gambiarras

✓ Sobre os testes * 1/1

Selecione as alternativas verdadeiras

- ☒ Teste unitário testa um pedaço muito isolado de código ✓
- ☐ Teste unitário testa uma rota por completo, desde a requisição até a resposta
- ☒ Devemos configurar um banco de dados apenas para testes na aplicação ✓
- ☐ Pra que testes? Podemos testar todas telas nós mesmos



Google Forms

