

PROGRAMAÇÃO PARA DISPOSITIVOS MÓVEIS - Resultados

1 Ao desenvolver um aplicativo que irá mostrar todo o catálogo de produtos de uma empresa em 3-D e será necessário realizar uma animação desses produtos, qual é o melhor tipo de linguagem a ser escolhida?

- ☒ a A linguagem nativa pela sua performance em aplicativo pesados.
- ☐ b A linguagem Híbrido-Nativo apesar de ter um código para cada ambiente.
- ☐ c A linguagem Híbrido-WEB pela sua performance.
- ☐ d A linguagem Híbrido-Nativo por ter um único código para diversas plataformas.
- ☐ e A linguagem Híbrido-WEB pela sua facilidade de desenvolviment

Pontuação: 1

2 Ao desenvolver um aplicativo MVP, mas que seja necessário executar nos sistemas operacionais Android e iOS, qual o melhor tipo de linguagem?

- ☐ a Linguagem Nativa utilizando o *Swift* e o *Kotlin*.
- ☐ b Linguagem Nativa utilizando o *Javascript*.
- ☒ c React Native com o *Expo CLI*.
- ☐ d Linguagem Nativa utilizando o *Java* e o *Objective-C*.
- ☐ e React Native com o *React Native CLI*.


Pontuação: 1

3 Para montarmos uma tela igual abaixo, como deveria estar o nosso código utilizando a ferramenta *Flexbox*?



Contextualização para acessibilidade: essa imagem exibe um aplicativo de celular com o primeiro bloco ocupando a maior parte da tela, dois blocos em vermelho e azul ocupando o mesmo tamanho e, por último, um bloco em preto ocupando um espaço maior que o bloco azul e menor que o bloco verde.

a container: {
 flex: 2,
 backgroundColor: 'green',
},
container2: {
 flex: 1,
 backgroundColor: 'red',
},
container3: {
 flex: 1,
 backgroundColor: 'blue',
},
container4: {
 flex: 2,
 backgroundColor: 'black',
},

 **b** container: {
 flex: 3,
 backgroundColor: 'green',
},
container2: {
 flex: 1,
 backgroundColor: 'red',
},
container3: {
 flex: 1,
 backgroundColor: 'blue',
},
container4: {
 flex: 2,
 backgroundColor: 'black',
},

c container: {
 flex: 3,
 backgroundColor: 'green',
},
container2: {
 flex: 2,
 backgroundColor: 'red',
},
container3: {
 flex: 2,
 backgroundColor: 'blue',
},
container4: {
 flex: 2,
 backgroundColor: 'black',
},

```
d container: {  
  flex: 1,  
  backgroundColor: 'green',  
},  
container2: {  
  flex: 1,  
  backgroundColor: 'red',  
},  
container3: {  
  flex: 1,  
  backgroundColor: 'blue',  
},  
container4: {  
  flex: 2,  
  backgroundColor: 'black',  
},  
  
e container: {  
  flex: 1,  
  backgroundColor: 'green',  
},  
container2: {  
  flex: 1,  
  backgroundColor: 'red',  
},  
container3: {  
  flex: 1,  
  backgroundColor: 'blue',  
},  
container4: {  
  flex: 3,  
  backgroundColor: 'black',  
},
```

Pontuação: 1

- 4 Uma das grandes diferenças do *React* para as outras linguagens são os *states*. Utilizados em diversas áreas do código, quais as suas principais funções?
- a Armazena a instância de um objeto para enviar a outro componente.
 - b Armazenar uma informação durante todo ciclo de vida do componente, mas não renderiza a tela quando seu valor é alterado.
 - c Armazena as informações somente enquanto o componente realiza a renderização.
 - ☒ d Armazenar uma informação durante todo ciclo de vida do componente e realizar uma nova renderização quando seu valor é alterado.
 - e Receber aviso sobre as mudanças dos valores de uma determinada variável.

Pontuação: 1

- 5 Você precisa desenvolver um aplicativo com menu lateral com dois itens e, dentro desses itens, teremos algumas telas que abrirão de acordo com a informação que você tocará. Para isso, utilizaremos quais componentes?

- a** Para realizar essa navegação de telas, usaremos o componente *Screens* para organizar os menus e, dentro deles, criaremos funções com os componentes *Stacks* para fazer a pilha de telas.
- b** Para realizar essa navegação de telas, usaremos o componente *Drawer* para organizar os menus e, dentro deles, criaremos funções com os componentes *Screens* para fazer a pilha de telas.
- c** Para realizar essa navegação de telas, usaremos o componente *Menu* para organizar os menus e, dentro deles, criaremos funções com os componentes *Screens* para fazer a pilha de telas.
- d** Para realizar essa navegação de conjunto de telas, usaremos o componente *Menu* para organizar os menus e, dentro deles, criaremos funções com os componentes *Stacks* para fazer a pilha de telas.
- ☒ **e** Para realizar essa navegação de telas, usaremos o componente *Drawer* para organizar os menus e, dentro deles, criaremos funções com os componentes *Stacks* para fazer a pilha de telas.

Pontuação: 1

- 6** O *React native* permite que seja enviado ao usuário um *push notification* com uma informação relevante para executar alguma funcionalidade do aplicativo, para fins de marketing ou retenção de clientes. Com isso, para enviarmos uma notificação ao *device*, qual o fluxo precisamos seguir:
- a** Solicitamos a autorização de envio de notificações e nosso servidor envia a mensagem e *token* para os sites de envio de notificação (expo, firebase ou APNS).
 - b** Solicitamos a autorização de envio de notificações, coletamos o *token*, enviamos o *token* ao nosso servidor e nosso servidor envia a mensagem e *token* diretamente para o celular.
 - ☒ **c** Solicitamos a autorização de envio de notificações, obtemos o *token* e enviamos uma mensagem diretamente para o celular.
 - d** Solicitamos a autorização de envio de notificações, coletamos o *token*, enviamos o *token* ao nosso servidor e nosso servidor envia a mensagem e *token* para os sites de envio de notificação (expo, firebase ou APNS).
 - e** Obtemos o *token* e enviamos uma mensagem diretamente para o celular.

Pontuação: 1

- 7** Escreva sobre as vantagens do uso do React Navigation no desenvolvimento de aplicativos em React Native.

Vantagens: O React Navigation facilita a navegação e troca de informação em aplicações web, navegação entre telas e componentes, permite ao desenvolvedor personalizar a navegação entre telas e menus através de gavetas, abas ou pilhas. Possui os modos de navegação por drawer, tabs e outros. O conceito principal do React Navigation é o stack ou pilha, onde o primeiro a entrar e o último a sair. É compatível com o Redux permitindo a construção de aplicativos consistentes e escalonáveis pelo controle do estado da navegação. É compatível com diversas plataformas: Android, iOS e web.

Conceito: Certo - Pontuação: 4

Explicação:

O React Native é um conjunto de componentes importantes para realizar a gestão de telas e menus em aplicativos em React Native, viabilizando a navegação de aplicativos com componentes como o Tab, Drawer e outros.

Legenda:

 Alternativa correta

 Resposta do aluno

Pontuação total: **9**