



Anterior

Dica do Professor

Próximo

Na prática



Exercícios

📄 Respostas enviadas em: 15/04/2021 17:14

1. WebApps estão no mercado *mobile* para preencher uma lacuna deixada pelos aplicativos nativos de dispositivos móveis no que diz respeito à portabilidade e à facilidade de desenvolvimento. Sobre WebApps, *site* responsivo e aplicativos nativos, assinale a alternativa correta:

☐ A. *Site* responsivo e WebApp são a mesma coisa, mas o WebApp permite criar atalhos.

Por que esta resposta não é correta?

WebApps são aplicativos que ficam disponíveis *off-line* e interagem com componentes nativos dos dispositivos. Aplicativos nativos são acessíveis apenas no sistema operacional para o qual foram desenvolvidos (por exemplo, no Android). Além disso, não são acessíveis em navegador. *Sites* responsivos não são a mesma coisa que os WebApps, sendo que estes têm características particulares. Por exemplo, os WebApps são acessíveis *off-line*, e os *sites* responsivos não.

☐ B. *Sites* responsivos e WebApps não são acessíveis *off-line*.

Por que esta resposta não é correta?

WebApps são aplicativos que ficam disponíveis *off-line* e interagem com componentes nativos dos dispositivos. Aplicativos nativos são acessíveis apenas no sistema operacional para o qual foram desenvolvidos (por exemplo, no Android). Além disso, não são acessíveis em navegador. *Sites* responsivos não são a mesma coisa que os WebApps, sendo que estes têm características particulares. Por exemplo, os WebApps são acessíveis *off-line*, e os *sites* responsivos não.

☐ C. WebApps se assemelham a aplicativos, mas não interagem com elementos nativos dos dispositivos (ex: câmera e GPS).

Por que esta resposta não é correta?

WebApps são aplicativos que ficam disponíveis *off-line* e interagem com componentes nativos dos dispositivos. Aplicativos nativos são acessíveis apenas no sistema operacional para o qual foram desenvolvidos (por exemplo, no Android). Além disso, não são acessíveis em navegador. *Sites* responsivos não são a mesma coisa que os WebApps, sendo que estes têm características particulares. Por exemplo, os WebApps são acessíveis *off-line*, e os *sites* responsivos não.

☐ D. Aplicativos nativos são acessíveis em qualquer navegador e em qualquer tamanho de tela.

Por que esta resposta não é correta?

WebApps são aplicativos que ficam disponíveis *off-line* e interagem com componentes nativos dos dispositivos. Aplicativos nativos são acessíveis apenas no sistema operacional para o qual foram desenvolvidos (por exemplo, no Android). Além disso, não são acessíveis em navegador. *Sites* responsivos não são a mesma coisa que os WebApps, sendo que estes têm características particulares. Por exemplo, os WebApps são acessíveis *off-line*, e os *sites* responsivos não.

Você acertou!

☒ E. WebApps são aplicativos que são disponíveis *off-line* e interagem com componentes nativos dos dispositivos.



Por que esta resposta é a correta?

WebApps são aplicativos que ficam disponíveis *off-line* e interagem com componentes nativos dos dispositivos. Aplicativos nativos são acessíveis apenas no sistema operacional para o qual foram desenvolvidos (por exemplo, no Android). Além disso, não são acessíveis em navegador. *Sites* responsivos não são a mesma coisa que os WebApps, sendo que estes têm características particulares. Por exemplo, os WebApps são acessíveis *off-line*, e os *sites* responsivos não.

1 de 5 perguntas

PRÓXIMA



Anterior

Dica do Professor

Próximo

Na prática



Exercícios

Respostas enviadas em: 15/04/2021 17:14

2. Um recurso que pode ser adicionado ao seu aplicativo *web* é *push notification*. Esse tipo de notificação está disponível na maioria dos aplicativos comerciais. Sobre essas notificações, assinale a alternativa correta:

- ☐ A. Um exemplo de *push notification* são as mensagens exibidas para confirmar ou cancelar uma ação, por exemplo, ao fechar o aplicativo.

Por que esta resposta não é correta?

Push notification são mensagens que o usuário do aplicativo recebe sem que ele tenha feito nenhuma ação para isso. As mensagens de confirmação e cancelamento são exemplos de *alerte* e não de *push notification*. Atualmente, é possível adicionar *push notification* aos PWAs; não é preciso utilizar nenhum tipo de emulador. As *push notifications* estão disponíveis em todos os sistemas operacionais. Um exemplo de *push notification* são notificações de chegada de mensagens, porque elas chegam sem o usuário programá-las, são exibidas na tela de bloqueio e, quando clica nelas, é direcionado para o aplicativo de origem.

- ☐ B. *Push notification* são mensagens agendadas pelo usuário que, posteriormente, são exibidas, como, por exemplo, um despertador.

Por que esta resposta não é correta?

Push notification são mensagens que o usuário do aplicativo recebe sem que ele tenha feito nenhuma ação para isso. As mensagens de confirmação e cancelamento são exemplos de *alerte* e não de *push notification*. Atualmente, é possível adicionar *push notification* aos PWAs; não é preciso utilizar nenhum tipo de emulador. As *push notifications* estão disponíveis em todos os sistemas operacionais. Um exemplo de *push notification* são notificações de chegada de mensagens, porque elas chegam sem o usuário programá-las, são exibidas na tela de bloqueio e, quando clica nelas, é direcionado para o aplicativo de origem.

- ☐ C. É possível inserir *push notification* em PWA. Contudo, para isso, é necessário fazer uso de um emulador.

Por que esta resposta não é correta?

Push notification são mensagens que o usuário do aplicativo recebe sem que ele tenha feito nenhuma ação para isso. As mensagens de confirmação e cancelamento são exemplos de *alerte* e não de *push notification*. Atualmente, é possível adicionar *push notification* aos PWAs; não é preciso utilizar nenhum tipo de emulador. As *push notifications* estão disponíveis em todos os sistemas operacionais. Um exemplo de *push notification* são notificações de chegada de mensagens, porque elas chegam sem o usuário programá-las, são exibidas na tela de bloqueio e, quando clica nelas, é direcionado para o aplicativo de origem.

- ☐ D. *Push notification* é um recurso exclusivo do sistema operacional Android.

Por que esta resposta não é correta?

Push notification são mensagens que o usuário do aplicativo recebe sem que ele tenha feito nenhuma ação para isso. As mensagens de confirmação e cancelamento são exemplos de *alerte* e não de *push notification*. Atualmente, é possível adicionar *push notification* aos PWAs; não é preciso utilizar nenhum tipo de emulador. As *push notifications* estão disponíveis em todos os sistemas operacionais. Um exemplo de *push notification* são notificações de chegada de mensagens, porque elas chegam sem o usuário programá-las, são exibidas na tela de bloqueio e, quando clica nelas, é direcionado para o aplicativo de origem.

Você acertou!

- ☒ E. Um exemplo de *push notification* são notificações de chegada de mensagens.



Por que esta resposta é a correta?

Push notification são mensagens que o usuário do aplicativo recebe sem que ele tenha feito nenhuma ação para isso. As mensagens de confirmação e cancelamento são exemplos de *alerte* e não de *push notification*. Atualmente, é possível adicionar *push notification* aos PWAs; não é preciso utilizar nenhum tipo de emulador. As *push notifications* estão disponíveis em todos os sistemas operacionais. Um exemplo de *push notification* são notificações de chegada de mensagens, porque elas chegam sem o usuário programá-las, são exibidas na tela de bloqueio e, quando clica nelas, é direcionado para o aplicativo de origem.

2 de 5 perguntas

< VOLTAR

PRÓXIMA >



Anterior

Dica do Professor

Próximo

Na prática



Exercícios

Respostas enviadas em: 15/04/2021 17:14

3. Um *site* responsivo já construído pode ser utilizado como porta de entrada para o desenvolvimento de um WebApp. Para isso, algumas ações precisam ser tomadas. Sobre essas ações, assinale a alternativa correta:

- ☐ A. É necessário criar um arquivo *manifest.json*, que é usado para disponibilizar o aplicativo nas lojas, por exemplo, no Google Play.

Por que esta resposta não é correta?

O que precisa ser inserido no *index* da aplicação é uma referência ao arquivo *service worker*. Além disso, o arquivo utilizado para, antes de fazer um *request*, buscar as informações no *cache*, para que seja feito um *request* apenas quando não existe no *cache*, também é o *service worker*. Já o arquivo *manifest.json* é usado para configurações iniciais do aplicativo como, por exemplo, a definição do ícone do aplicativo e a permissão para que o aplicativo seja instalado e, consequentemente, com a exibição de uma mensagem pra isso. Esse arquivo não tem relação com a possibilidade de disponibilização do aplicativo no Google Play.

- ☐ B. É necessário criar um arquivo *service worker* para que seja exibida uma mensagem para o usuário instalar o aplicativo no dispositivo.

Por que esta resposta não é correta?

O que precisa ser inserido no *index* da aplicação é uma referência ao arquivo *service worker*. Além disso, o arquivo utilizado para, antes de fazer um *request*, buscar as informações no *cache*, para que seja feito um *request* apenas quando não existe no *cache*, também é o *service worker*. Já o arquivo *manifest.json* é usado para configurações iniciais do aplicativo como, por exemplo, a definição do ícone do aplicativo e a permissão para que o aplicativo seja instalado e, consequentemente, com a exibição de uma mensagem pra isso. Esse arquivo não tem relação com a possibilidade de disponibilização do aplicativo no Google Play.

Resposta correta

- ☒ C. O ícone que será utilizado pelo aplicativo é informado no arquivo *manifest.json*.

Por que esta resposta é a correta?

O que precisa ser inserido no *index* da aplicação é uma referência ao arquivo *service worker*. Além disso, o arquivo utilizado para, antes de fazer um *request*, buscar as informações no *cache*, para que seja feito um *request* apenas quando não existe no *cache*, também é o *service worker*. Já o arquivo *manifest.json* é usado para configurações iniciais do aplicativo como, por exemplo, a definição do ícone do aplicativo e a permissão para que o aplicativo seja instalado e, consequentemente, com a exibição de uma mensagem pra isso. Esse arquivo não tem relação com a possibilidade de disponibilização do aplicativo no Google Play.

Você não acertou!

- ☐ D. O arquivo *manifest.json* é utilizado para, antes de fazer um *request*, buscar as informações no *cache* para que seja feito um *request* apenas quando não existe no *cache*.

Por que esta resposta não é correta?

O que precisa ser inserido no *index* da aplicação é uma referência ao arquivo *service worker*. Além disso, o arquivo utilizado para, antes de fazer um *request*, buscar as informações no *cache*, para que seja feito um *request* apenas quando não existe no *cache*, também é o *service worker*. Já o arquivo *manifest.json* é usado para configurações iniciais do aplicativo como, por exemplo, a definição do ícone do aplicativo e a permissão para que o aplicativo seja instalado e, consequentemente, com a exibição de uma mensagem pra isso. Esse arquivo não tem relação com a possibilidade de disponibilização do aplicativo no Google Play.

- ☐ E. É preciso, por fim, associar o *manifest.json* no *site* por meio da inserção de uma referência ao *manifest.json* no *index.htm* da sua aplicação.

Por que esta resposta não é correta?

O que precisa ser inserido no *index* da aplicação é uma referência ao arquivo *service worker*. Além disso, o arquivo utilizado para, antes de fazer um *request*, buscar as informações no *cache*, para que seja feito um *request* apenas quando não existe no *cache*, também é o *service worker*. Já o arquivo *manifest.json* é usado para configurações iniciais do aplicativo como, por exemplo, a definição do ícone do aplicativo e a permissão para que o aplicativo seja instalado e, consequentemente, com a exibição de uma mensagem pra isso. Esse arquivo não tem relação com a possibilidade de disponibilização do aplicativo no Google Play.



Anterior

Dica do Professor

Próximo

Na prática



Exercícios

Respostas enviadas em: 15/04/2021 17:14

4. Um WebApp também pode ser chamado de PWA. Porém, para ser considerado um PWA, o *site* precisa ter algumas características bem específicas. Sobre elas, assinale a alternativa correta:

- ☐ A. Progressivo: deve ser adaptável a qualquer computador *desktop*, celular e *tablet*, independente do tamanho de tela.

Por que esta resposta não é correta?

As principais características de um PWA são:

Progressivo: deve funcionar para qualquer navegador.

Responsivo: deve ser adaptável a qualquer computador *desktop*, celular e *tablet*, independente do tamanho de tela.

Seguro: deve utilizar o protocolo HTTPS para evitar invasões e garantir que o conteúdo não seja adulterado.

Reenvolvente: facilita o reengajamento com recursos como notificações *push*.

Linkável: deve permitir compartilhamento por URL; não requer instalação complexa.

- ☐ B. Responsivo: deve funcionar para qualquer navegador.

Por que esta resposta não é correta?

As principais características de um PWA são:

Progressivo: deve funcionar para qualquer navegador.

Responsivo: deve ser adaptável a qualquer computador *desktop*, celular e *tablet*, independente do tamanho de tela.

Seguro: deve utilizar o protocolo HTTPS para evitar invasões e garantir que o conteúdo não seja adulterado.

Reenvolvente: facilita o reengajamento com recursos como notificações *push*.

Linkável: deve permitir compartilhamento por URL; não requer instalação complexa.

Resposta correta

- ☒ C. Reenvolvente: facilita o reengajamento com recursos nativos do dispositivo como, por exemplo, notificações *push*.

Por que esta resposta é a correta?

As principais características de um PWA são:

Progressivo: deve funcionar para qualquer navegador.

Responsivo: deve ser adaptável a qualquer computador *desktop*, celular e *tablet*, independente do tamanho de tela.

Seguro: deve utilizar o protocolo HTTPS para evitar invasões e garantir que o conteúdo não seja adulterado.

Reenvolvente: facilita o reengajamento com recursos como notificações *push*.

Linkável: deve permitir compartilhamento por URL; não requer instalação complexa.

- ☐ D. Localizável: deve permitir compartilhamento por URL; não requer instalação complexa.

Por que esta resposta não é correta?

As principais características de um PWA são:

Progressivo: deve funcionar para qualquer navegador.

Responsivo: deve ser adaptável a qualquer computador *desktop*, celular e *tablet*, independente do tamanho de tela.

Seguro: deve utilizar o protocolo HTTPS para evitar invasões e garantir que o conteúdo não seja adulterado.

Reenvolvente: facilita o reengajamento com recursos como notificações *push*.

Linkável: deve permitir compartilhamento por URL; não requer instalação complexa.

Você não acertou!

- ☐ E. Seguro: deve utilizar o protocolo HTTP para evitar invasões e garantir que o conteúdo não seja adulterado.

Por que esta resposta não é correta?

As principais características de um PWA são:

Progressivo: deve funcionar para qualquer navegador.

Responsivo: deve ser adaptável a qualquer computador *desktop*, celular e *tablet*, independente do tamanho de tela.

Seguro: deve utilizar o protocolo HTTPS para evitar invasões e garantir que o conteúdo não seja adulterado.

Reenvolvente: facilita o reengajamento com recursos como notificações *push*.

Linkável: deve permitir compartilhamento por URL; não requer instalação complexa.



Anterior

Dica do Professor

Próximo

Na prática



Exercícios

Respostas enviadas em: 15/04/2021 17:14

5. Uma das principais características de um PWA é estar sempre disponível mesmo *off-line*; para que isso seja possível, é necessário efetuar a programação de um componente específico no seu PWA, chamado de Service Worker. Assinale a alternativa que contém a descrição correta de alguma característica desse componente:

Você acertou!



A. Service Worker é um *script* executado pelo navegador em segundo plano.



Por que esta resposta é a correta?

Service Worker é um *script* executado pelo navegador em segundo plano. É preciso associar o Service Worker ao *site* por meio da inserção dele no arquivo *index.htm*. O *cache* de um aplicativo é versionado pela variável *CACHE_NAME*. O Service Worker permite interceptar e tratar solicitações de rede, respondendo com um *cache* quando o aplicativo tiver. Atualmente, os Service Workers têm recursos de notificações de *push*.



B. Os Service Workers não incluem recursos de notificações *push*.

Por que esta resposta não é correta?

Service Worker é um *script* executado pelo navegador em segundo plano. É preciso associar o Service Worker ao *site* por meio da inserção dele no arquivo *index.htm*. O *cache* de um aplicativo é versionado pela variável *CACHE_NAME*. O Service Worker permite interceptar e tratar solicitações de rede, respondendo com um *cache* quando o aplicativo tiver. Atualmente, os Service Workers têm recursos de notificações de *push*.



C. O Service Worker atua para que não seja possível interceptar e tratar solicitações de rede, respondendo com *cache*.

Por que esta resposta não é correta?

Service Worker é um *script* executado pelo navegador em segundo plano. É preciso associar o Service Worker ao *site* por meio da inserção dele no arquivo *index.htm*. O *cache* de um aplicativo é versionado pela variável *CACHE_NAME*. O Service Worker permite interceptar e tratar solicitações de rede, respondendo com um *cache* quando o aplicativo tiver. Atualmente, os Service Workers têm recursos de notificações de *push*.



D. O Service Worker tem uma versão chamada *CACHE_VERSION*, que é utilizada para versionar o *cache* do aplicativo.

Por que esta resposta não é correta?

Service Worker é um *script* executado pelo navegador em segundo plano. É preciso associar o Service Worker ao *site* por meio da inserção dele no arquivo *index.htm*. O *cache* de um aplicativo é versionado pela variável *CACHE_NAME*. O Service Worker permite interceptar e tratar solicitações de rede, respondendo com um *cache* quando o aplicativo tiver. Atualmente, os Service Workers têm recursos de notificações de *push*.



E. O Service Worker precisa ser linkado ao *site* no arquivo *propert.xml*.

Por que esta resposta não é correta?

Service Worker é um *script* executado pelo navegador em segundo plano. É preciso associar o Service Worker ao *site* por meio da inserção dele no arquivo *index.htm*. O *cache* de um aplicativo é versionado pela variável *CACHE_NAME*. O Service Worker permite interceptar e tratar solicitações de rede, respondendo com um *cache* quando o aplicativo tiver. Atualmente, os Service Workers têm recursos de notificações de *push*.

5 de 5 perguntas

< VOLTAR