

Questionário de Conhecimentos em Desenvolvimento Flutter e Tópicos Relacionados

Instruções: Escolha a alternativa que melhor completa ou responde a cada pergunta.

1. Processos em Segundo Plano: Qual é o modelo do Dart para multithreading, utilizado para executar código Dart em segundo plano, que difere de uma thread convencional por não compartilhar memória com o programa principal? A) Threads Compartilhadas B) Workers Assíncronos C) **Isolates** D) Goroutines

2. Processos em Segundo Plano: Qual plugin é mencionado por permitir o processamento persistente em segundo plano no Flutter, mantendo as tarefas agendadas mesmo após o aplicativo ser reiniciado ou o sistema ser reinicializado? A) BackgroundFetch B) **WorkManager** C) AsyncScheduler D) PersistentTasks

3. Comunicação com WebSockets: Ao contrário das requisições HTTP normais, que tipo de comunicação os WebSockets permitem com um servidor, sem a necessidade de polling? A) Comunicação unidirecional B) Polling de sentido único C) **Comunicação bidirecional** D) Apenas comunicação de broadcast

4. Comunicação com WebSockets: Qual pacote do Flutter fornece a classe `WebSocketChannel`, que permite tanto ouvir mensagens do servidor quanto enviar mensagens para ele, facilitando a conexão com um servidor WebSocket? A) `http_client` B) `socket_io_client` C) **`web_socket_channel`** D) `stream_socket`

5. Comunicação com WebSockets: No Dart, qual classe, fundamental no pacote `dart:async`, pode entregar muitos eventos assíncronos ao longo do tempo, diferentemente de `Future` que retorna uma única resposta assíncrona? A) Queue B) **Stream** C) Iterable D) Channel

6. Comparativo Dio vs. HTTP (Uso Geral): Segundo alguns desenvolvedores na comunidade Flutter, para necessidades básicas de requisições API, qual pacote é considerado mais simples e adequado? A) Dio B) **http** C) Chopper D) Retrofit

7. Comparativo Dio vs. HTTP (Funcionalidades Avançadas): Qual recurso avançado é frequentemente citado como uma das principais vantagens do pacote **Dio**, permitindo a execução de código customizado a cada requisição para lidar com autenticação, tratamento de erros ou logging? A) Cache embutido B) Cancelamento de requisições C) **Interceptors** D) Compressão de dados

8. Comparativo Dio vs. HTTP (Cancelamento e Protocolos): Qual dos pacotes de requisições HTTP para Flutter oferece **cancelamento de requisições** e suporte a HTTP/2, mas é notado por não suportar HTTP/3 (QUIC)? A) http B) dart:io C) **Dio** D) fetch

9. Aplicações Híbridas: Aplicações móveis híbridas são instaladas diretamente no dispositivo, mas são desenvolvidas com tecnologias que são geralmente direcionadas para qual ambiente? A) Código nativo (Swift/Kotlin) B) **Tecnologias web (como HTML5,**

Angular) C) Linguagem de montagem D) Linguagens proprietárias específicas do dispositivo

10. Persistência de Dados Offline (Contexto TOTVS): No projeto de estágio na TOTVS, qual plugin foi utilizado para o armazenamento eficiente de variáveis do tipo Chave-Valor com acesso rápido, sendo crucial para a persistência de informações leves e o funcionamento da aplicação? A) SQLite B) Hive C) **Shared Preferences** D) Realm

11. Persistência de Dados Offline (Contexto TOTVS): Qual mecanismo de banco de dados foi empregado no projeto de estágio da TOTVS para a persistência interna dos dados provenientes do ERP, especificamente na versão 2.3.0? A) Firestore B) Realm C) PostgreSQL D) **SQFlite**

12. Sincronização de Dados (Contexto TOTVS): Para otimizar a sincronização e garantir que o ERP retorne apenas os dados que foram alterados ou adicionados desde a última operação, qual contexto (`/contexto`) foi implementado dentro de cada API encarregada de fornecer os dados das entidades? A) `/delta` B) `/update` C) `/diff` D) `/changes`

13. Criação Dinâmica de Formulários (Contexto TOTVS): No projeto de estágio da TOTVS, qual algoritmo de ordenação foi utilizado para permitir a disposição personalizada e dinâmica dos campos do formulário de Pedido de Venda, baseando-se em critérios estabelecidos pelo administrador? A) Bubble Sort B) Merge Sort C) Insertion Sort D) **QuickSort**

14. Arquitetura e Compilação Flutter: Comparado ao motor Hermes do React Native, qual é uma vantagem fundamental do motor Flutter em relação à compilação e vulnerabilidades, destacando a compilação AOT (Ahead-Of-Time)? A) Utiliza exclusivamente a compilação Just-In-Time (JIT). B) Possui vulnerabilidades previamente relatadas. C) **Oferece compilação Ahead-Of-Time (AOT) para código de máquina e não possui vulnerabilidades previamente relatadas.** D) Depende de motores JavaScript nativos.

15. Segurança Flutter - Ofuscação: Qual é uma limitação do ofuscador Dart embutido do Flutter em relação à proteção de informações sensíveis dentro do aplicativo? A) Não consegue randomizar nomes de símbolos. B) Não impede a visualização da estrutura da classe. C) Fornece ofuscação de strings para segredos no aplicativo. D) **Não fornece ofuscação de strings, o que pode potencialmente expor informações sensíveis.**

16. Segurança Flutter - TLS Pinning: Por que as técnicas padrão para estabelecer um proxy ou interceptar o tráfego de rede são frequentemente ineficazes contra a biblioteca TLS e os componentes de rede do Flutter? A) Flutter depende da pilha TLS nativa do dispositivo. B) Flutter respeita as configurações de proxy definidas pelo sistema operacional. C) **A biblioteca TLS e os componentes de rede do Flutter são integrados diretamente ao motor Flutter.** D) Flutter utiliza um enclave seguro separado para todo o tráfego de rede.

17. Uso do Pacote HTTP: Ao fazer múltiplas requisições HTTP para o mesmo servidor usando o pacote `http`, qual é a prática recomendada para manter uma conexão persistente aberta e evitar a criação de requisições avulsas repetidamente? A) Fazer chamadas

`http.get` individuais para cada requisição. B) **Utilizar uma instância de `http.Client`.**
C) Implementar um `StreamedRequest` customizado. D) Mudar para o pacote `Dio`.

18. Pacote HTTP - Retentativas: Qual classe do `package:http/retry.dart` permite envolver um `http.Client` para retentar transparentemente requisições falhas, por padrão retentando status code 503? A) `HttpClientRetry` B) `RequestRetryer` C) **`RetryClient`** D) `NetworkRetry`

19. WebSockets (Tempo Real): De acordo com o vídeo "Realtime Applications with Sockets - Introduction in Sockets with Dart", qual é um dos principais benefícios de usar uma conexão de socket em vez de uma requisição HTTP para aplicações modernas? A) Reduz o tamanho total da transferência de dados. B) Elimina a necessidade de um servidor. C) **Permite comunicação em tempo real, mantendo uma conexão estável que notifica o cliente quando algo muda.** D) Suporta apenas comunicação unidirecional do cliente para o servidor.

20. Integração de API Flutter: Em uma aplicação Flutter, ao buscar dados de uma API, o que as palavras-chave `async` e `await` garantem em relação à resposta? A) O código prosseguirá imediatamente sem esperar pela resposta. B) A resposta é tratada de forma síncrona. C) **O código não prosseguirá até que a resposta seja recebida, tratando-a de forma assíncrona.** D) A resposta é sempre armazenada em cache localmente.

Gabarito

1. C
2. B
3. C
4. C
5. B
6. B
7. C
8. C
9. B
10. C
11. D
12. C
13. D
14. C
15. D
16. C
17. B
18. C
19. C
20. C

