

Questionário: Fundamentos de Dart e Flutter

Instruções: Escolha a opção que melhor responde a cada pergunta, com base nas informações dos materiais fornecidos.

- 1. Qual é a principal função da linguagem Dart no ecossistema Flutter?** A) Criar bancos de dados para aplicativos. B) Gerenciar servidores web e APIs. C) Ser a linguagem de programação criada pelo Google e utilizada para escrever o Flutter, permitindo a criação de aplicativos nativos de alta performance. D) Desenvolver sistemas operacionais.

- 2. Que tipo de experiência o Dart proporciona ao desenvolvedor e ao usuário final, de acordo com as fontes?** A) Uma experiência complexa para o desenvolvedor e um aplicativo interpretado para o usuário. B) Uma incrível experiência para o desenvolvedor e um aplicativo nativo de alta performance para o usuário final. C) Uma experiência de baixo desempenho para ambos, mas com maior flexibilidade. D) Uma experiência limitada, focada apenas em desenvolvimento web.

- 3. Se você já tem experiência com linguagens "C-style" (como C#, Javascript), como a Dart é considerada em termos de aprendizado?** A) Extremamente difícil de aprender, pois sua sintaxe é completamente diferente. B) Muito amigável e fácil de aprender, devido à sua sintaxe C-style. C) Moderadamente desafiadora, exigindo um profundo conhecimento de estruturas de dados. D) Uma linguagem legada com pouca documentação.

- 4. Quais das seguintes características descrevem corretamente a linguagem Dart?** A) Não tipada, antiga e sem suporte a funções de alta ordem. B) Tipada (opcionalmente), moderna (com generics, funções, interfaces e mixins), e que pode ser compilada e interpretada. C) Apenas interpretada e focada exclusivamente em desenvolvimento de backend. D) Fortemente tipada (sempre explícita), sem capacidade de compilação.

- 5. Qual é o propósito principal do `dartpad.dev`?** A) Um editor de texto offline para desenvolvimento Dart. B) Um IDE online para Dart e Flutter, que permite escrever, executar e compartilhar códigos diretamente em um navegador. C) Um repositório de pacotes para a linguagem Dart. D) Uma ferramenta para depuração exclusiva de aplicativos Flutter em dispositivos móveis.

- 6. No Dart, como a tipagem "opcional" funciona e qual benefício ela oferece?** A) Permite declarar variáveis sem tipo, mas desativa os benefícios da tipagem estática. B) Não existe tipagem opcional; todas as variáveis devem ter seu tipo explicitamente declarado. C) Através da inferência, o Dart identifica o tipo da variável, garantindo todos os benefícios da tipagem estática. D) Significa que o Dart é uma linguagem fracamente tipada, similar ao JavaScript em termos de segurança de tipo.

- 7. Qual é a função principal do `main()` em um código Dart?** A) É opcional e serve apenas para definir variáveis globais. B) É a função necessária para executar um código Dart. C) Define a interface de usuário do aplicativo. D) Gerencia o tratamento de exceções no programa.

- 8. De acordo com as fontes, qual é a diferença crucial entre Dart e JavaScript em relação ao tratamento de tipos ao tentar realizar uma operação com uma String de forma incorreta?** A) Ambos lançam exceções, mas JavaScript oferece mais detalhes. B) Dart lança uma exceção, enquanto no JavaScript, embora o comportamento seja incorreto, o programa continua funcionando. C) JavaScript lança uma exceção, enquanto Dart ignora o erro. D) Nenhuma das duas linguagens possui tratamento de exceções para tipos.

- 9. Qual palavra-chave é usada no Dart para declarar uma propriedade de classe que, uma vez atribuída durante a criação da instância, não pode ter seu valor modificado (imutável)?** A) `var`. B) `const`. C) `static`. D) `final`.

- 10. Como se define uma propriedade privada em uma classe Dart?** A) Utilizando a palavra-chave `private` antes do nome da propriedade. B) Colocando um asterisco (*) antes do

nome da propriedade. C) Colocando um underscore (`_`) antes do nome da propriedade. D) Declarando a propriedade fora da classe.

11. No contexto do Null Safety do Dart, qual símbolo é utilizado para sinalizar que uma propriedade de uma classe pode ser nula? A) `#`. B) `!`. C) `?`. D) `@`.

12. Qual é o papel principal da pasta `lib` em um projeto Dart/Flutter recém-criado? A) Conter o arquivo principal para rodar a aplicação (`main.dart`). B) Armazenar os testes da aplicação. C) É onde se criam os módulos, modelos, controllers e serviços que formam toda a aplicação. D) Definir os recursos externos e bibliotecas do projeto.

13. Qual das seguintes opções não é listada como um tipo de dado básico ou especial suportado pelo Dart, de acordo com as fontes? A) `boolean`. B) `strings`. C) `rune`. D) `char`.

14. Qual a principal vantagem de utilizar Flutter, conforme descrito nas fontes? A) É executado em uma WebView, garantindo compatibilidade universal. B) Gera código verdadeiramente nativo tanto para Android quanto para iOS a partir de um único código-base. C) Utiliza uma ponte para acessar componentes nativos, similar ao React Native. D) Requer um código-base separado para cada plataforma.

15. Qual ferramenta é recomendada para instalar o Flutter SDK e é um sistema de controle de versão do qual o Flutter depende? A) Node.js. B) Maven. C) Git. D) Docker.

16. Onde a comunidade Dart/Flutter publica e você pode encontrar bibliotecas e extensões para uso em seus projetos? A) npmjs.com. B) PyPI (Python Package Index). C) pub.dev. D) rubygems.org.

17. Qual é o comando de terminal utilizado para criar um novo projeto Dart? A) `dart new project_name`. B) `dart generate app_name`. C) `dart init project_name`. D) `dart create meu_app` (ou `dart create nome_do_projeto`).

18. Qual é uma das desvantagens do Dart mencionada nas fontes? A) É uma linguagem antiga e desatualizada. B) Possui uma comunidade muito grande e com excesso de padronização. C) É uma linguagem relativamente nova, com comunidade menor e menor adoção por empresas. D) Não é open-source e é muito cara para usar.

19. Em um projeto Dart, qual arquivo é responsável por dizer ao Dart onde encontrar recursos como imagens, fontes e módulos externos, além de listar as bibliotecas do `pub.dev` a serem utilizadas? A) `main.dart`. B) `README.md`. C) `pubspec.yaml`. D) `analysis_options.yaml`.

20. Como o Flutter consegue executar o mesmo aplicativo tanto em Android quanto em iOS, e qual o impacto disso na performance de animações? A) Ele executa em uma WebView, limitando a performance a 30 fps. B) Ele utiliza uma "ponte" para componentes nativos, o que causa latência nas animações. C) Ele renderiza os elementos do aplicativo (chamados Widgets) diretamente em um Canvas, o que permite executar animações em até 120 fps. D) Ele compila o código Dart para JavaScript, que é executado no navegador embutido, resultando em desempenho variável.

Gabarito

Aqui estão as respostas corretas para o questionário, com as respectivas citações das fontes:

1. **C)** Ser a linguagem de programação criada pelo Google e utilizada para escrever o Flutter, permitindo a criação de aplicativos nativos de alta performance.
2. **B)** Uma incrível experiência para o desenvolvedor e um aplicativo nativo de alta performance para o usuário final.

3. **B)** Muito amigável e fácil de aprender, devido à sua sintaxe C-style.
4. **B)** Tipada (opcionalmente), moderna (com generics, funções, interfaces e mixins), e que pode ser compilada e interpretada.
5. **B)** Um IDE online para Dart e Flutter, que permite escrever, executar e compartilhar códigos diretamente em um navegador.
6. **C)** Através da inferência, o Dart identifica o tipo da variável, garantindo todos os benefícios da tipagem estática.
7. **B)** É a função necessária para executar um código Dart.
8. **B)** Dart lança uma exceção, enquanto no JavaScript, embora o comportamento seja incorreto, o programa continua funcionando.
9. **D)** `final`.
10. **C)** Colocando um underscore (`_`) antes do nome da propriedade.
11. **C)** `?`.
12. **C)** É onde se criam os módulos, modelos, controllers e serviços que formam toda a aplicação.
13. **D)** `char`.
14. **B)** Gera código verdadeiramente nativo tanto para Android quanto para iOS a partir de um único código-base.
15. **C)** Git.
16. **C)** `pub.dev`.
17. **D)** `dart create meu_app` (ou `dart create nome_do_projeto`).
18. **C)** É uma linguagem relativamente nova, com comunidade menor e menor adoção por empresas.
19. **C)** `pubspec.yaml`.
20. **C)** Ele renderiza os elementos do aplicativo (chamados Widgets) diretamente em um Canvas, o que permite executar animações em até 120 fps.