

Introdução ao Flutter

Quanto a linguagem de programação usada

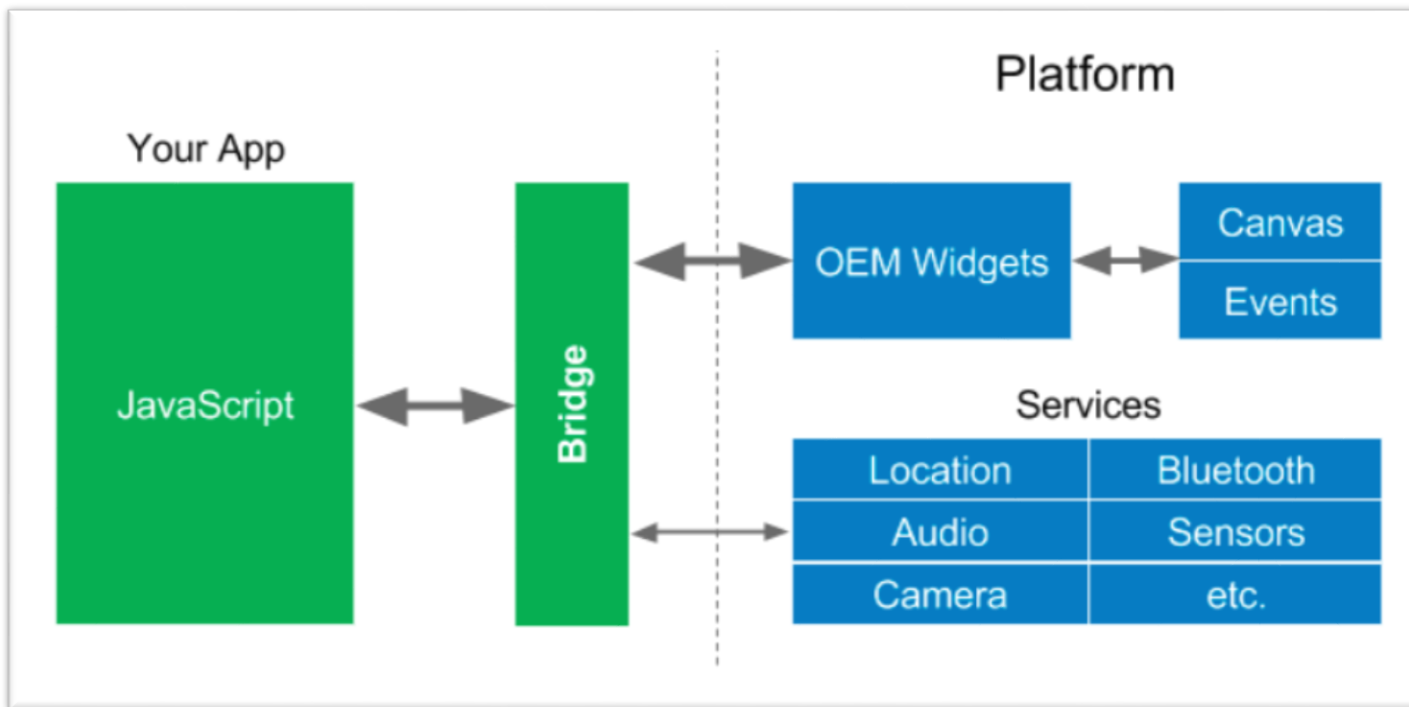
- Flutter utiliza Dart como linguagem de programação.
- Mas por que Dart? E não JavaScript?
 - Visto que outros frameworks híbridos usam o JavaScript.
- JavaScript usa compilação just-in-time (JIT).
 - Código compilado em tempo de execução
- Dart usa compilação ahead-of-time (AOT).
 - Código compilado em tempo de desenvolvimento, antes da execução.

Quanto a compilação

- JIT pode ser problema quando queremos desenvolver aplicações fluídas rodando animações com 60 fps.
 - Compilação em tempo de execução pode gerar gargalos de performance, e produzir "congelamentos" no uso da app.
- AOT faz toda compilação antes de execução, sendo assim, seu carregamento e fluidez com animações não sofrem do problema gerado pelo JIT.
 - Só isso já justificaria o uso do Dart!

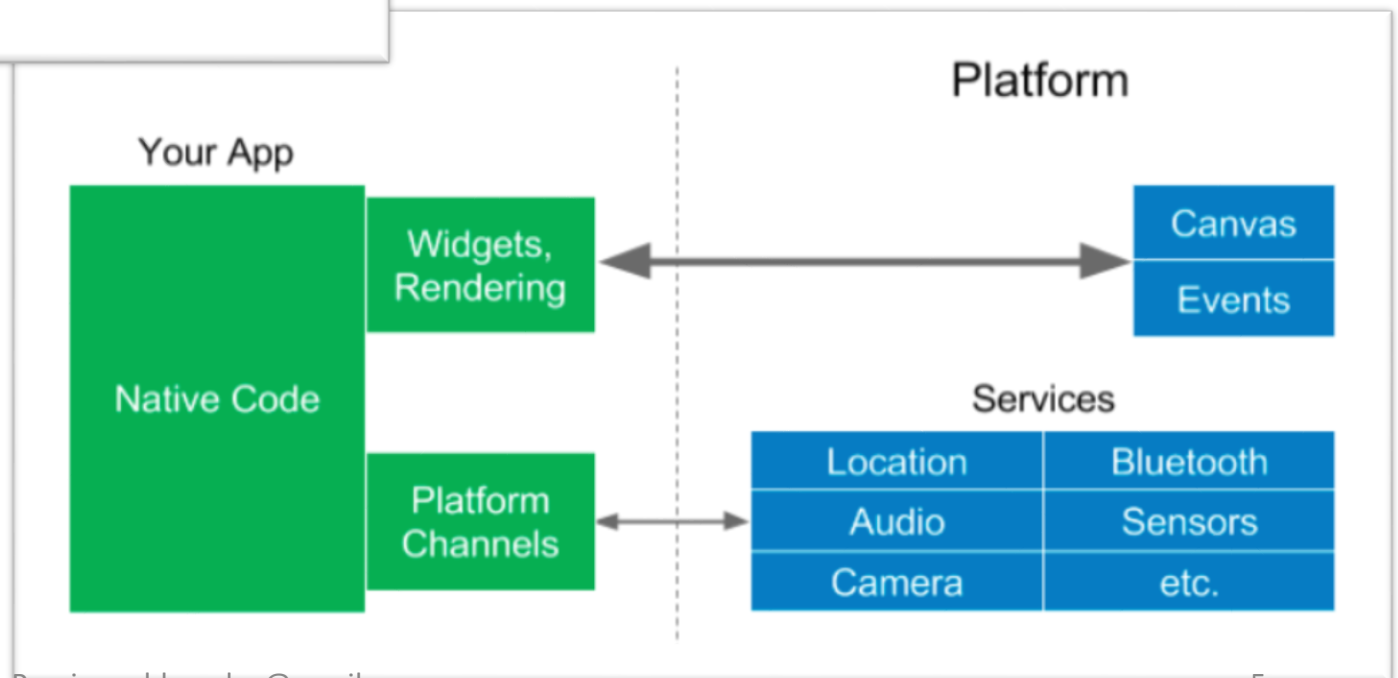
Quanto ao uso de Bridge

- React Native e similares usam o JavaScript como linguagem de programação.
- Basicamente o JavaScript se comunica com a plataforma nativa (vice-versa) através de uma implementação de uma camada chamada Bridge.
- O problema do Bridge é que ele sofre com tráfego de grande quantidade de dados.
- Embora existem cada vez mais melhorias das bridges sendo realizada pelos times de desenvolvedores de cada framework.



← React Native e similares

Flutter →

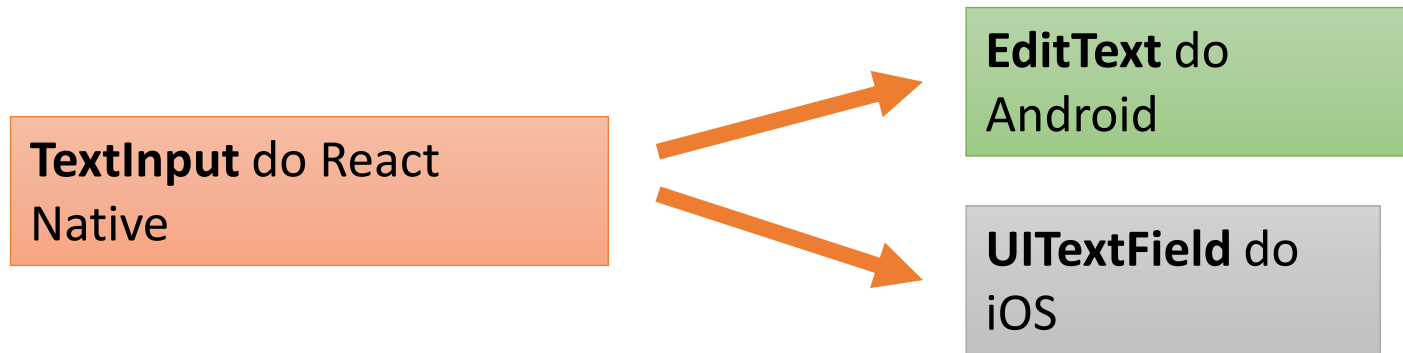


Quanto ao uso de Bridge

- O Bridge em algum momento pode se tornar um gargalo de performance em sua aplicação como por exemplo
 - Renderizar animações personalizadas,
 - Trafegar arquivos e dados JSON grandes ou
 - Quando temos muitos dados sendo exibidos em um mapa.
- No Flutter o código Dart é compilado para a plataforma nativa, não existe a necessidade de Bridge.
 - Problemas relacionados a gargalo de performance em sua aplicação são minimizados.

Quanto aos componentes de UI

- React Native e similares utilizam componentes de UI nativos.



- O time de desenvolvimento do Flutter decidiu não usar essa abordagem.
 - Em vez disso eles criaram um próprio conjunto de componentes de UI

Quanto aos componentes de UI

- Então quer dizer que um componente como uma caixa de texto tem seu próprio visual, não respeitando o visual da plataforma que está sendo executada?
 - **Sim e não!**
- O Flutter respeita o visual de cada plataforma com ajuda da implementação de dois temas:
 - Material Components que é o tema do Android
 - Cupertino que é o tema do iOS
- O Flutter permite utilizar os dois temas em cada plataforma, e nada impede que o desenvolvedor customize os componentes.

Quanto aos componentes de UI

- Flutter disponibiliza uma das melhores implementações do Material Design;
- Material Design é uma linguagem visual criada pelo Google;
- Objetivo do Material Design é proporcionar ao usuário no mundo digital a mesma percepção dos objetos no mundo real;



<https://olhardigital.com.br/noticia/saiba-o-que-e-e-quais-sao-as-vantagens-do-material-design/52910>

Quanto aos componentes de UI

- Os componentes de UI no Flutter são chamados de Widgets, esses são os componentes que permitem um usuário interagir com as funcionalidades do app.
- Tudo é um Widget no Flutter (esse é um lema que os desenvolvedores levaram muito a sério)
 - Botões, caixas de texto e menus;
 - Elementos de estilo como fonte, cores e temas;
 - Aspectos de layout como espaçamento e posicionamento;
 - Controle de gestos,
 - Até mesmo o próprio aplicativo.

Quanto aos componentes de UI

- Além do mais é possível executar uma aplicação Flutter em versões anteriores do Android e iOS com o mesmo visual sem precisar de libs de compatibilidade (muito comum no desenvolvimento nativo com Android)

Quanto as ferramentas de desenvolvimento

- Atualmente temos oficialmente um plugin para o IntelliJ, Android Studio e o Visual Studio Code.
- O plugin do IntelliJ e Android Studio é mais completo, pois possui o Widget Inspector que permite navegar na árvore de componentes da tela.
- As ferramentas apresentam a funcionalidade de Hot Reload, permite atualizar o código e a aplicação sem recompilar tudo novamente.
- O Flutter tem compilação AOT, compilado antes da execução, como é possível o Hot Reload? A resposta é simples!
 - O time de desenvolvimento sabiamente utiliza o JIT, compilação em tempo de execução, para que isso seja possível!

Anatomia de um App Flutter

