Flutter

Filipe Veber

- Área de TI: 2013
- SENAC - ADS: 2018
- Dev generalista

ZIPDEV

- San Diego - CA
- Staff recruiting

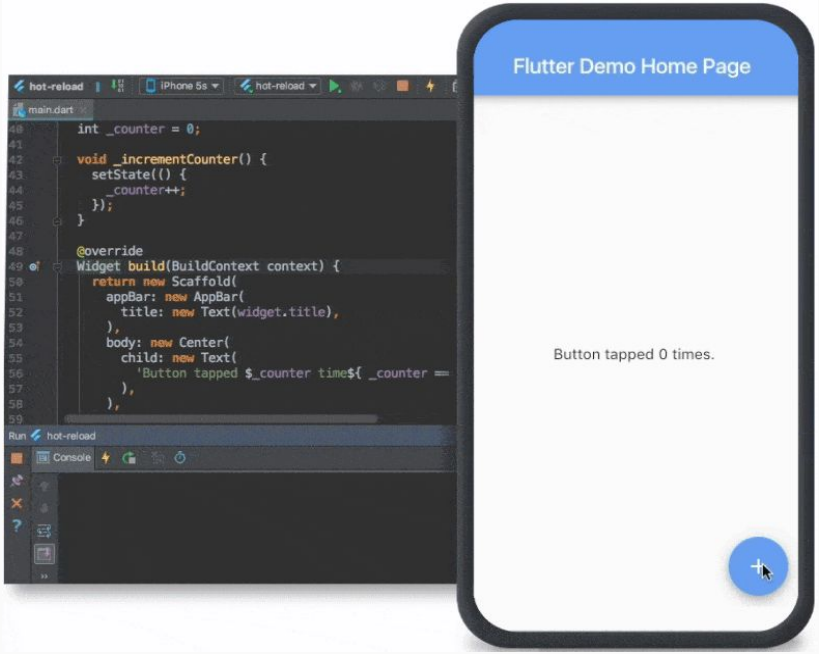
- ZipDevelopers: 109
- Active clients: 35
- Countries: 14

Flutter

“Flutter is Google’s UI toolkit for building beautiful, natively compiled applications for mobile, web, and desktop from a single codebase” - flutter.dev

Fast development

Paint your app to life in milliseconds with Stateful Hot Reload. Use a rich set of fully-customizable widgets to build native interfaces in minutes.



Expressive and Flexible UI

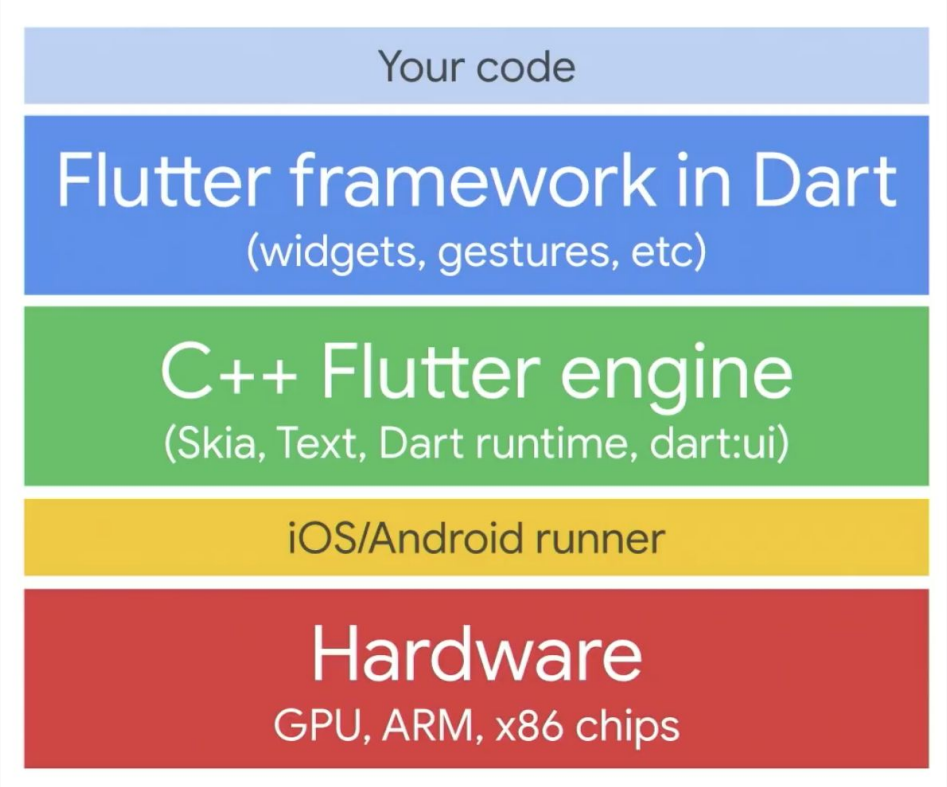
Quickly ship features with a focus on native end-user experiences. Layered architecture allows for full customization, which results in incredibly fast rendering and expressive and flexible designs.

Native performance

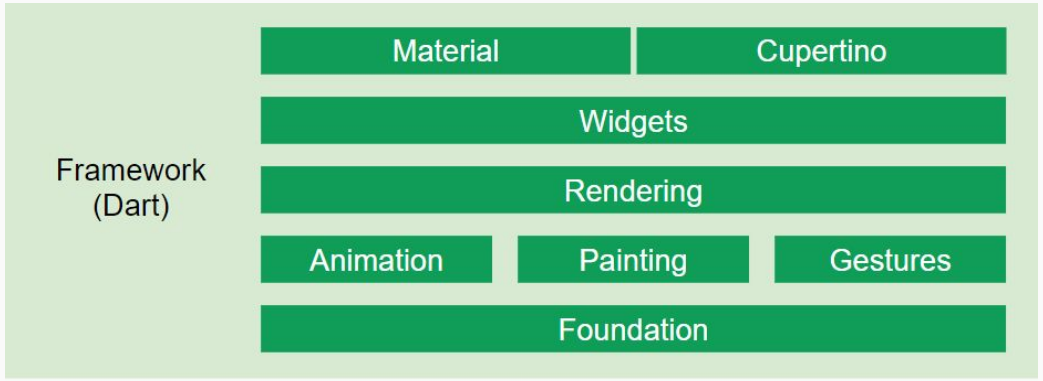
Flutter’s widgets incorporate all critical platform differences such as scrolling, navigation, icons and fonts, and your Flutter code is compiled to native ARM machine code using Dart's native compilers.

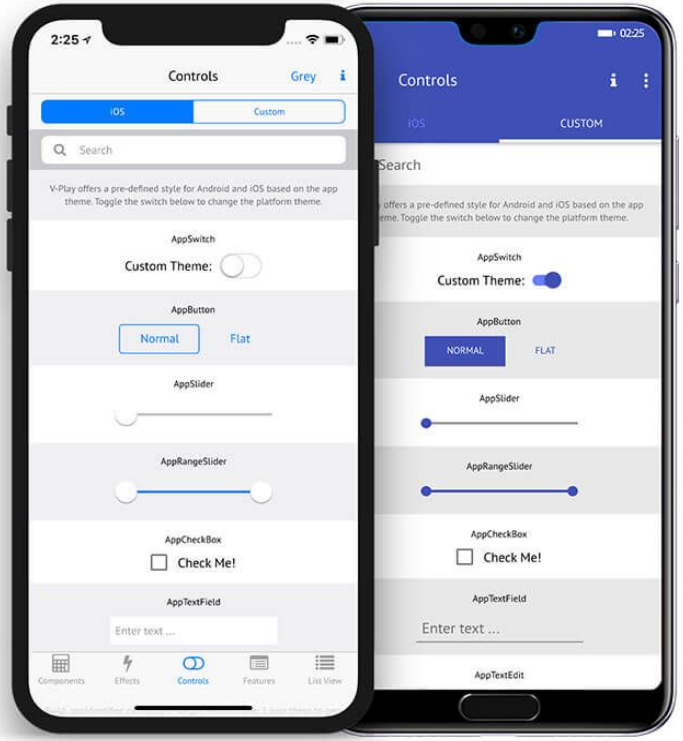

SO, ARE YOU
TELLING ME THAT
THE SAME CODE RUNS FOR
BOTH ANDROID AS WELL AS IOS?
imgflip.com

Architecture



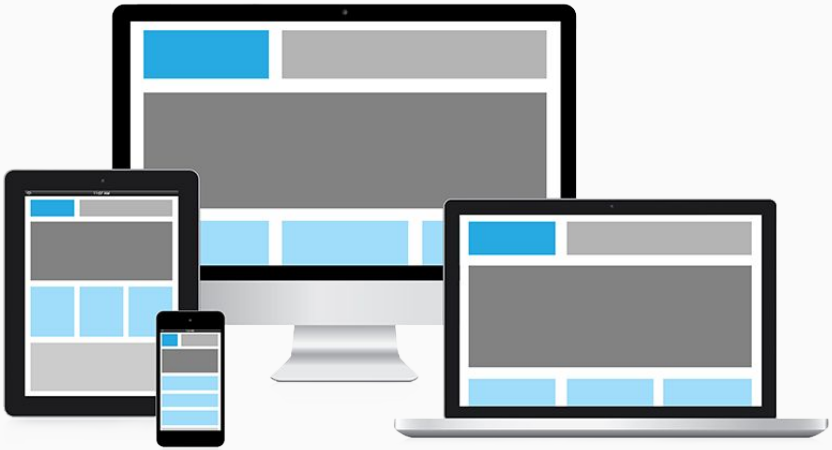
Architecture





Cross-platform

Cross-platform products are those you develop with a single programming language and a generator build native versions to each target platform.



Dart

What

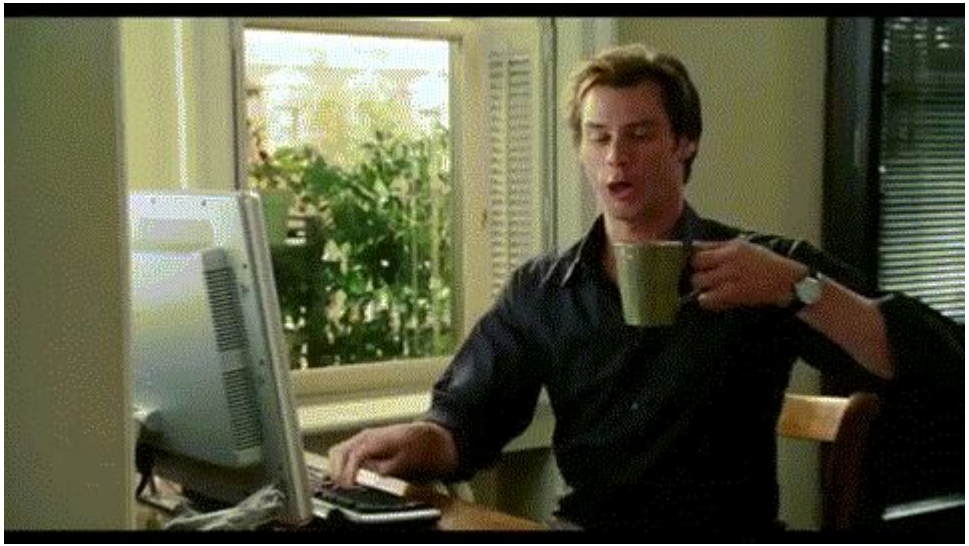
“Dart is a client-optimized language for fast apps on any platform” - dart.dev

Optimized for UI

Develop with a programming language specialized around the needs of user interface creation

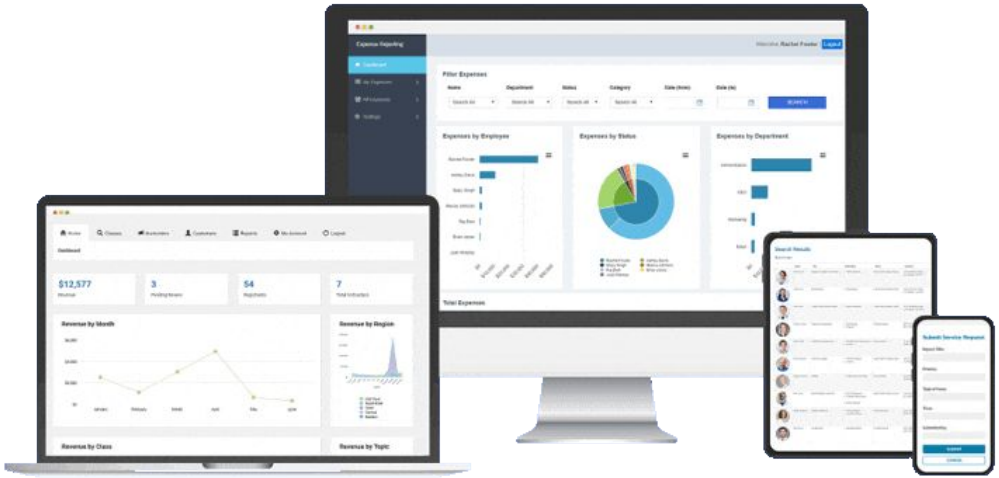
Productive development

Make changes iteratively: use hot reload to see the result instantly in your running app



Fast on all platforms

Compile to ARM & x64 machine code for mobile, desktop, and backend. Or compile to JavaScript for the web

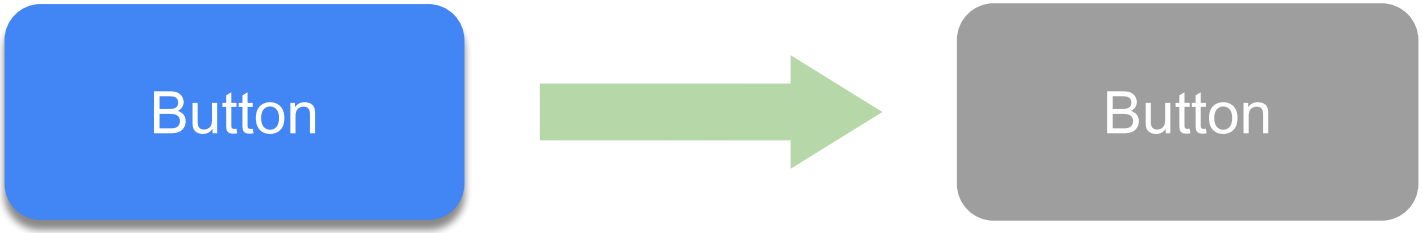


Widgets

What

“Widgets describe what their view should look like given their current configuration and state. When a widget’s state changes, the widget rebuilds its description...” - flutter.dev

What

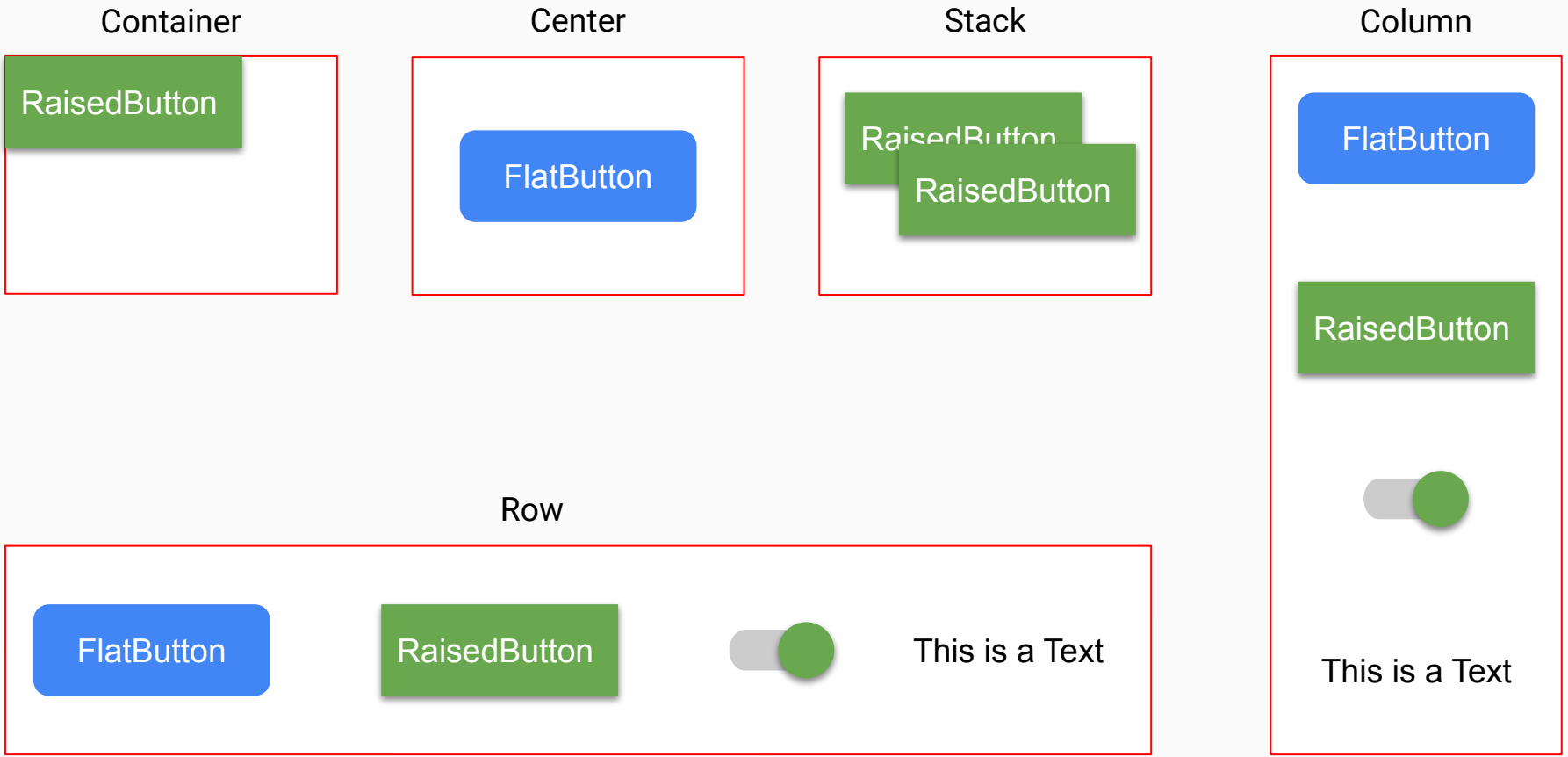


Descrição	Cor de fundo azul	
	Cor do texto branco	
	Texto Button	
Estado	Habilitado	

Descrição	Cor de fundo cinza	
	Cor do texto branco	
	Texto Button	
Estado	Desabilitado	


I SEE WIDGETS
EVERYWHERE

Layout Widgets x Interface Widgets



State management

What

“...everything that exists in memory when the app is running…” - flutter.dev

“Whatever data you need in order to rebuild your UI at any moment in time.” - flutter.dev

Ephemeral state

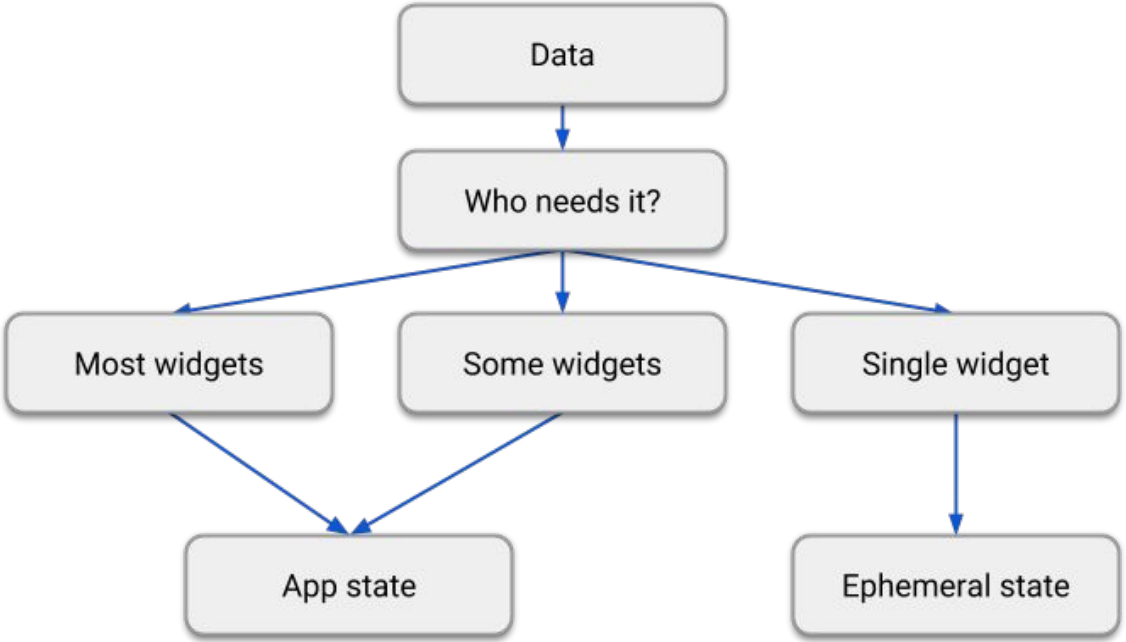
Ephemeral state (sometimes called UI state or local state) is the state you can neatly contain in a single widget.

- Current page in a PageView
- Current progress of a complex animation
- Current selected tab in a BottomNavigationBar
- An enabled / disabled button depending on a form

Global state

State that is not ephemeral, that you want to share across many parts of your app, and that you want to keep between user sessions, is what we call application state (sometimes also called shared state).

- User preferences
- Login info
- Notifications in a social networking app
- The shopping cart in an e-commerce app
- Read/unread state of articles in a news app


Data
Who needs it?
Most widgets
Some widgets
Single widget
App state
Ephemeral state

State managers

setState

Provider

BLoC

GetX

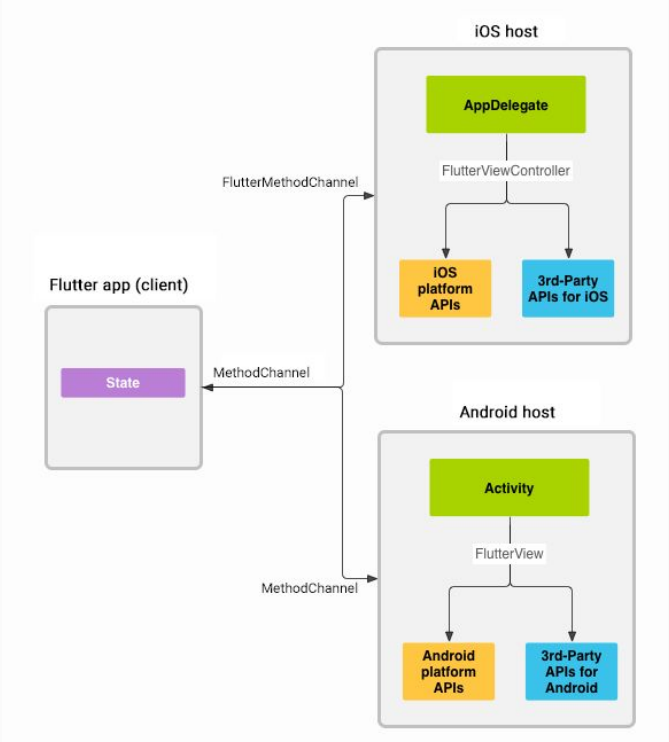
MobX

Cubit

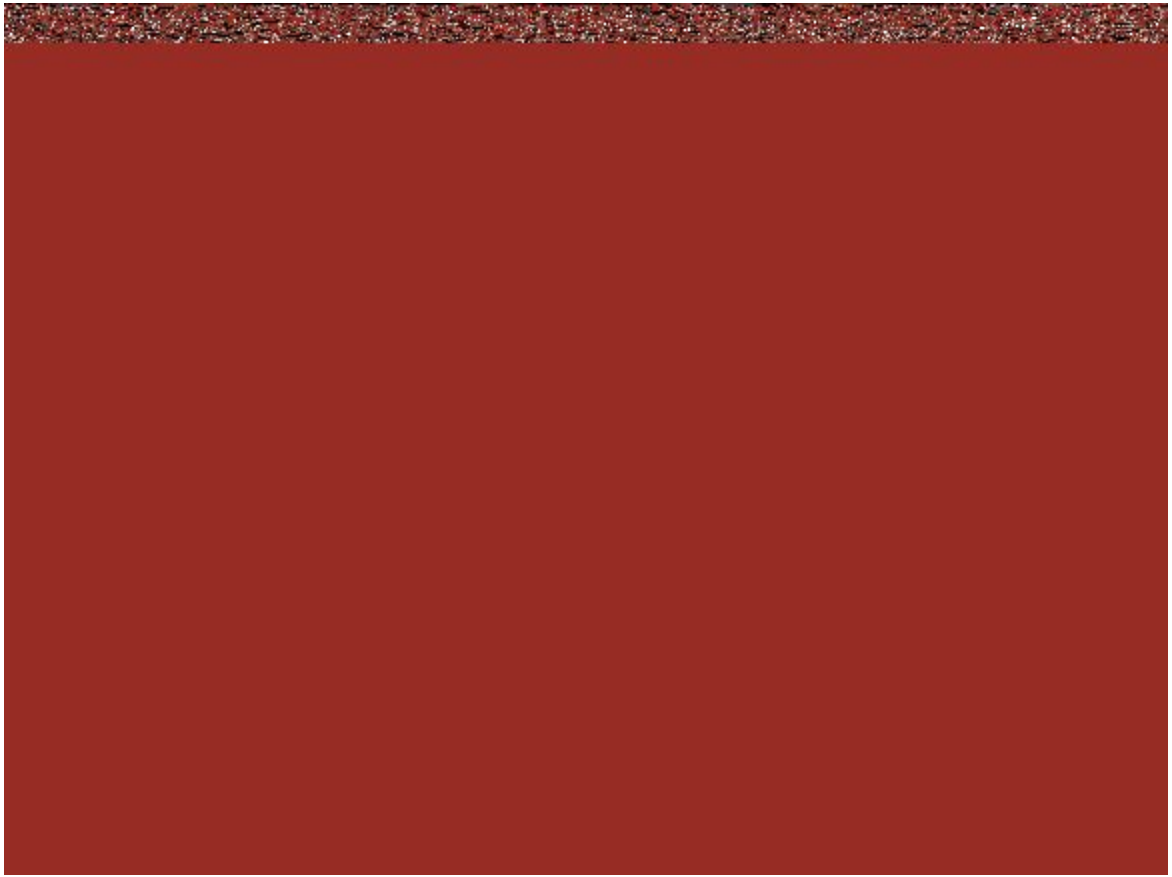
“Both types [local and global] have their place in any Flutter app, and the split between the two depends on your own preference and the complexity of the app.” - flutter.dev

Integration with native code

Platform-channels



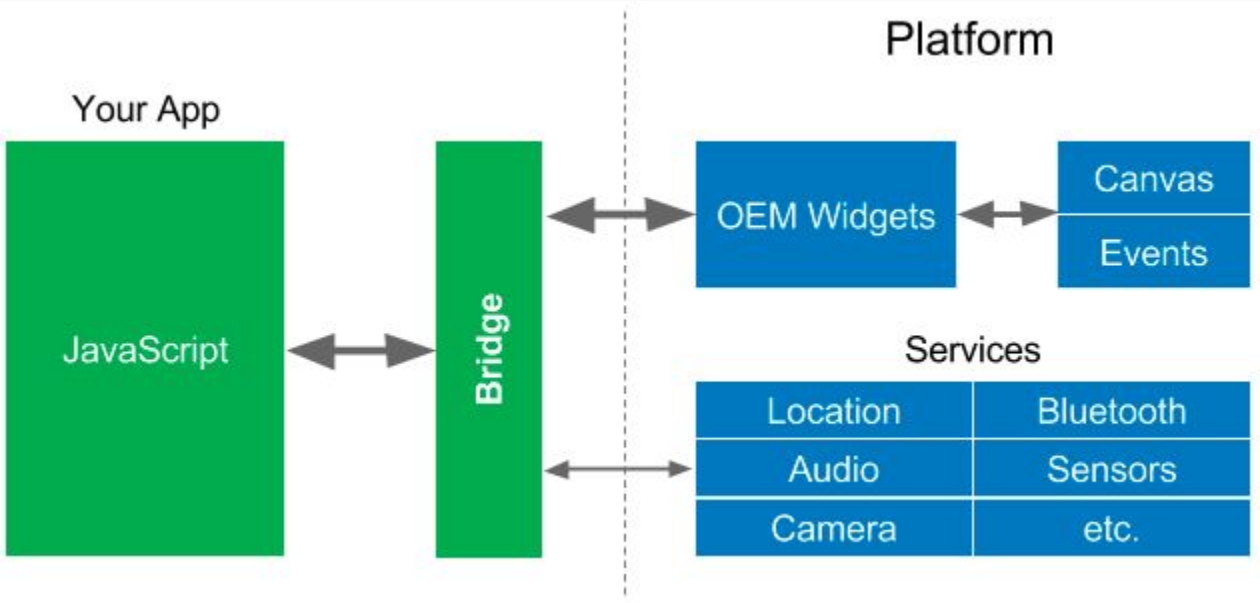
Frameworks comparison



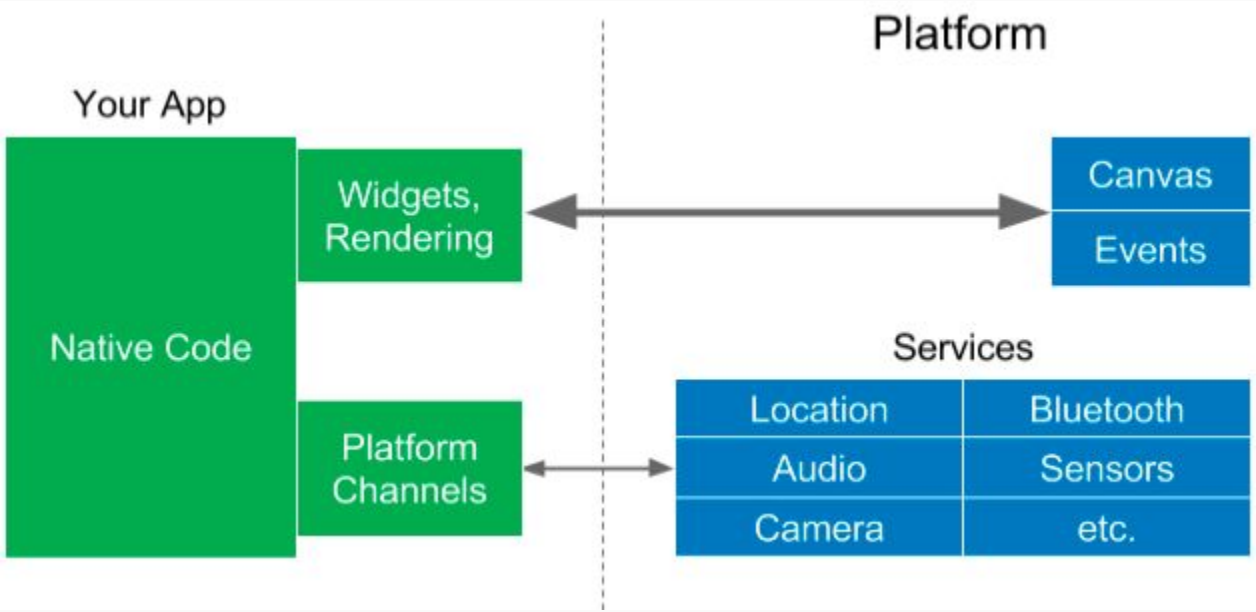
Technical features

Flutter	React Native	Ionic
Dart + Flutter	Javascript / React.js	Javascript
Compiled native app	Partially compiled native app	App inside a WebView
Don’t use native components from iOS/Android	Do use native components from iOS/Android	Don’t use native components from iOS/Android
Cross-Platform (Mobile, Web e Desktop)	Mobile	Cross-Platform (Mobile, Web e Desktop)
Made by Google	Made by Facebook	Made by Ionic

Other frameworks



Flutter

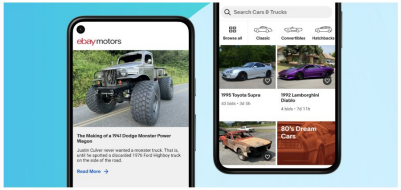

Platform
Your App
Native Code
Widgets,
Rendering
Platform
Channels
Canvas
Events
Services
Location
Bluetooth
Audio
Sensors
Camera
etc.


REACT NATIVE
XAMARIN
FLUTTER
imgflip.com

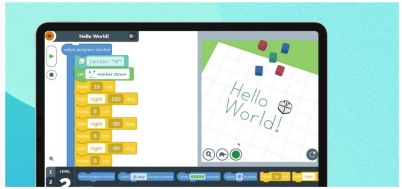
Showcase

Who is using Flutter?

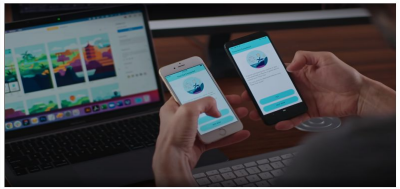
eBay



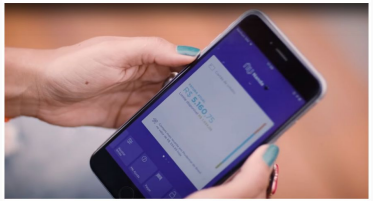
iRobot



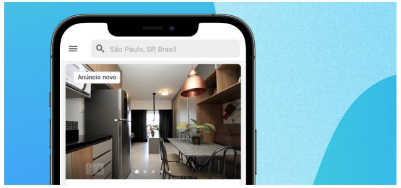
US Department of Veterans Affairs



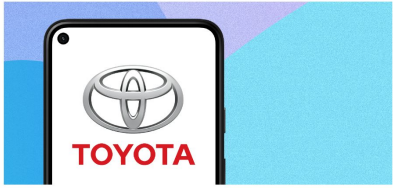
Nubank



QuintoAndar

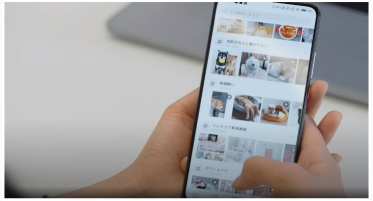


Toyota



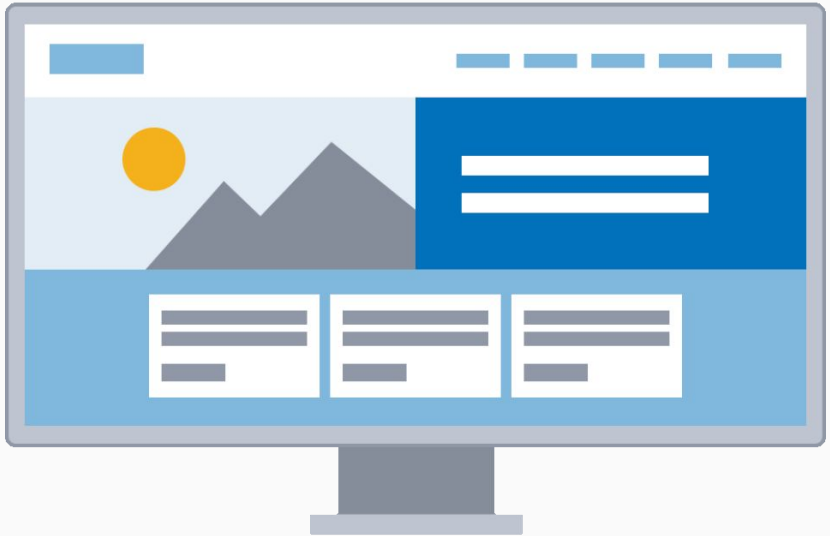
BMW

ByteDance



Responsive

“Responsive designs respond to changes in browser width by adjusting the placement of design elements to fit the available space.” - Ethan Marcotte



Pros

Any screen size, no matter the device

Search engine friendly

Standardized layout. Better user experience

Cons

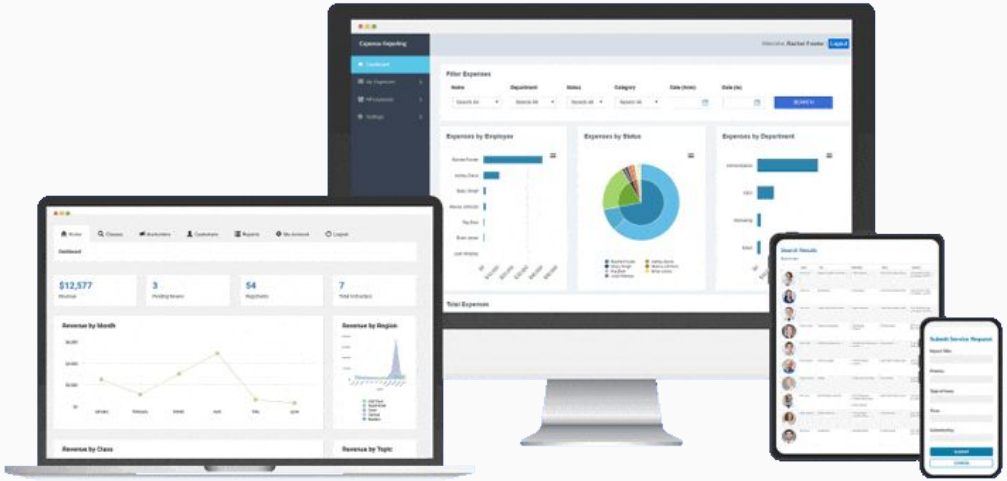
Guarantee eligible and accessible elements in different sizes

Requires more code than traditional websites, increasing development time

Risks of breaking the layout after updates to used browsers or frameworks

Adaptive

“The adaptive design has several fixed layout sizes. When the site detects the available space, it selects the most appropriate layout for the screen”



Pros	Cons
Perfect design to all devices’ browsers	Uncommon devices may break the design
Higher score in speed tests	Content duplication makes SEO difficult
More personalization capabilities, connected to using smart device options and adaptation	Design the same site multiple times

Responsive x Adaptive

Breakpoints

“Breakpoint is the “point” at which a website’s content and design will adapt in a certain way to provide the best possible user experience.”

Screens

320 px

480 px

760 px

960 px

1,200 px

1,600 px

Platforms

Web

Mobile

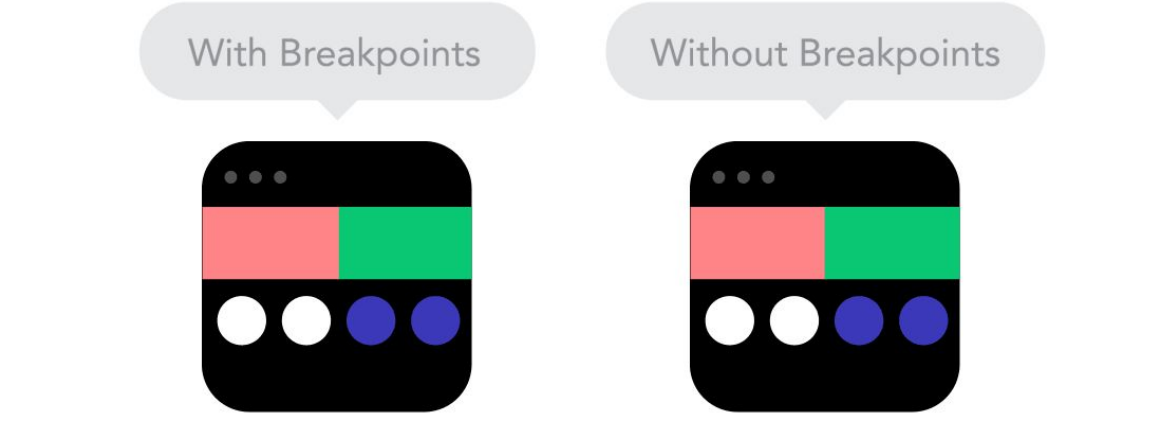
Desktop

Windows

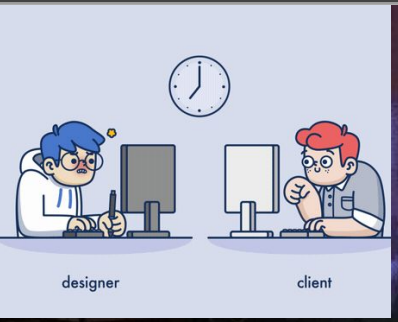
MacOS

Linux

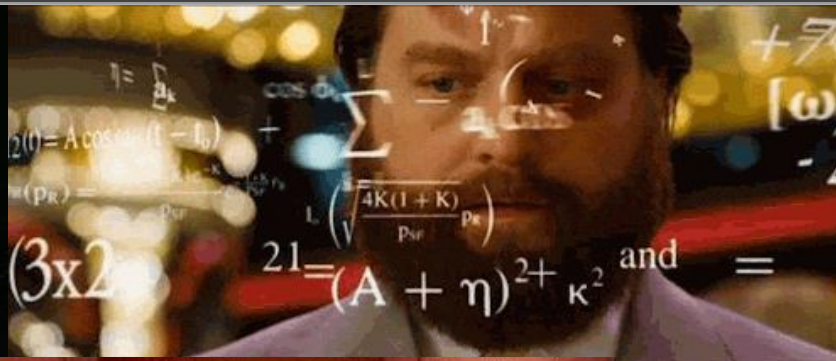
Breakpoints

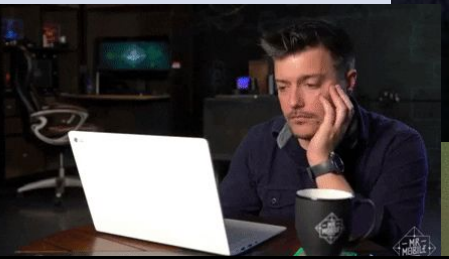


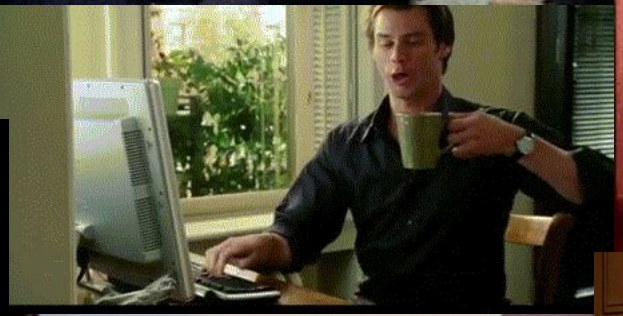
In the end...


designer
client


ME
PROJECT MANAGER


(3x2
21=(A + η)2+ κ2
and






Web Designers
What my friends think I do


Programmer
BUG


peacock

Referências

https://flutter.dev

https://dart.dev

https://docs.flutter.dev/development/ui/layout/adaptive-responsive

https://docs.flutter.dev/development/ui/layout/building-adaptive-apps

https://blog.codemagic.io/building-responsive-applications-with-flutter

https://www.springboard.com/blog/design/responsive-vs-adaptive-design

https://designshack.net/articles/ux-design/responsive-vs-adaptive-design/

https://www.interaction-design.org/literature/article/adaptive-vs-responsive-design

https://xd.adobe.com/ideas/process/ui-design/adaptive-design-vs-responsive-design/

Obrigado!

Salmos 136:1 - “Dêem graças ao Senhor, porque Ele é bom. O Seu amor dura para sempre!”

Utilização de breakpoint adaptativo de plataforma

```
Widget buildPhotosWidget(List<PhotoModel> items) {
  return LayoutBuilder(
    builder: (context, BoxConstraints constraints) {
      if (kIsWeb) {
        return buildGridView(items);
      }
      return buildListView(items);
    },
  );
}
```

Utilização de breakpoint adaptativo de plataforma e de tela

```dart
Widget buildPhotosWidget(List<PhotoModel> items) {
  return LayoutBuilder(
    builder: (context, BoxConstraints constraints) {
      if (kIsWeb) {
        if (constraints.maxWidth <= 980) {
          return buildListView(items);
        }
        return buildGridView(items);
      }
      return buildListView(items);
    },
  );
}
```

Utilização de breakpoint responsivo de tela

```
Widget buildGridView(List<PhotoModel> items) {
  return LayoutBuilder(
    builder: (_, constraints) {
      var columns = 6;
      double padding = 8;

      if (constraints.maxWidth <= 760) {
        columns = 2;
        padding = 18;
      }

      return GridView.builder(
        gridDelegate: SliverGridDelegateWithFixedCrossAxisCount(
            crossAxisCount: columns),
        itemBuilder: (context, index) {
          final photo = items[index];
          return Padding(
            padding: EdgeInsets.all(padding),
            child: buildGridTile(context, photo),
          );
        },
      );
    },
  );
}
```