Flutter

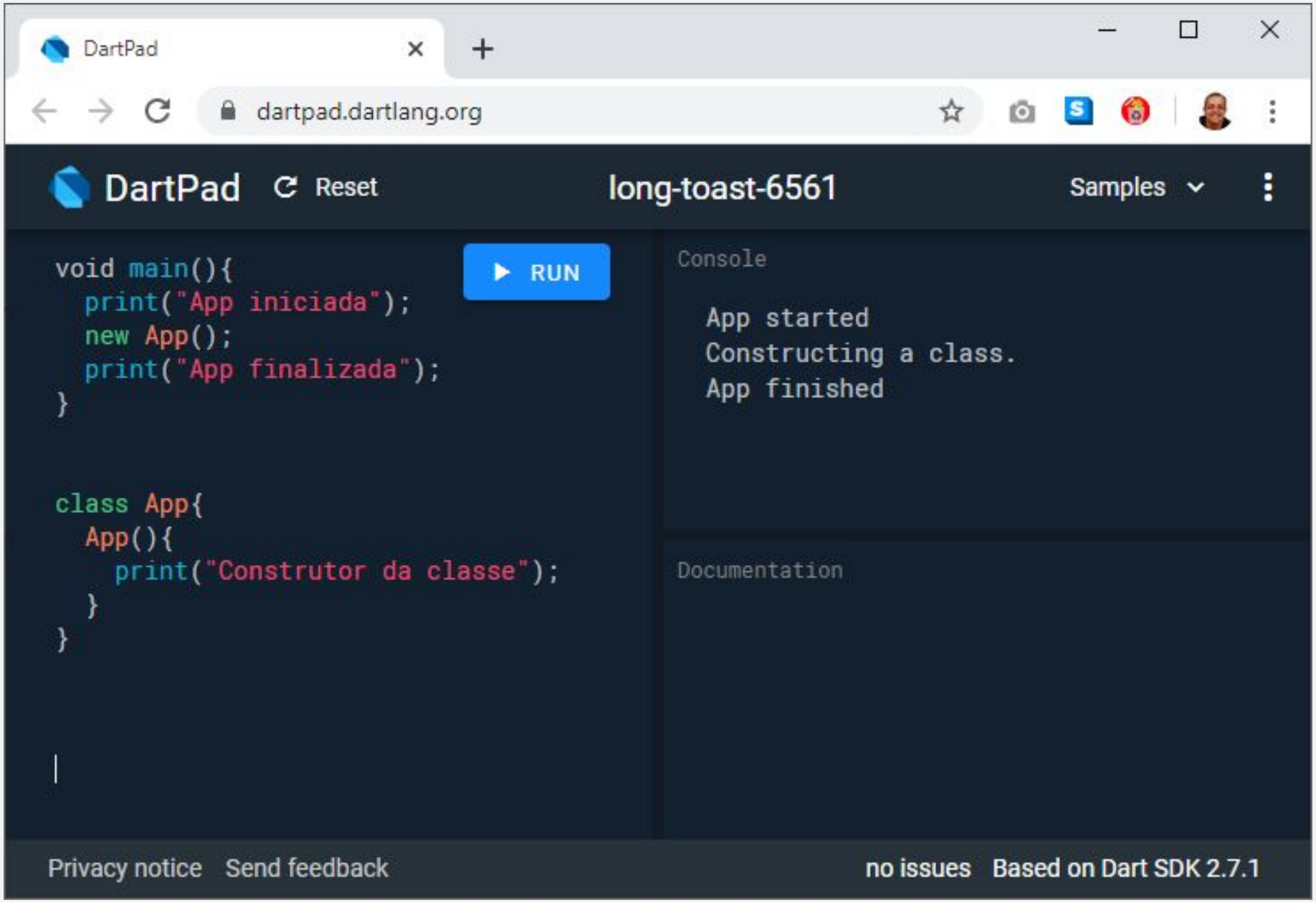
Aula 04

Prof. Dr. Rodrigo Plotze
rodrigoplotze@gmail.com

Roteiro

- DART
- Widgets
  - Value Widgets
  - Navigation Widgets
- Estrutura Básica

DART


DartPad
dartpad.dartlang.org
DartPad
Reset
long-toast-6561
Samples
RUN
void main(){
print("App iniciada");
new App();
print("App finalizada");
}
class App{
App(){
print("Construtor da classe");
}
}
Console
App started
Constructing a class.
App finished
Documentation
Privacy notice
Send feedback
no issues
Based on Dart SDK 2.7.1

dartpad.dev

Dart

- Declaração de variáveis
  - Tipo seguido do identificador:
  - A palavra reservada ***var*** declara uma variável sem um tipo específico, a qual recebe um tipo dinamicamente.
  - A palavra reservada ***dinamic*** declara uma variável do tipo dinâmico com um tipo opcional

```
var x;

    ...or...

<some specific type> x;
```

```
var x = "Mel  Brooks";
String x = "Mel  Brooks";
```

```
dynamic x = "Mel Books";
```

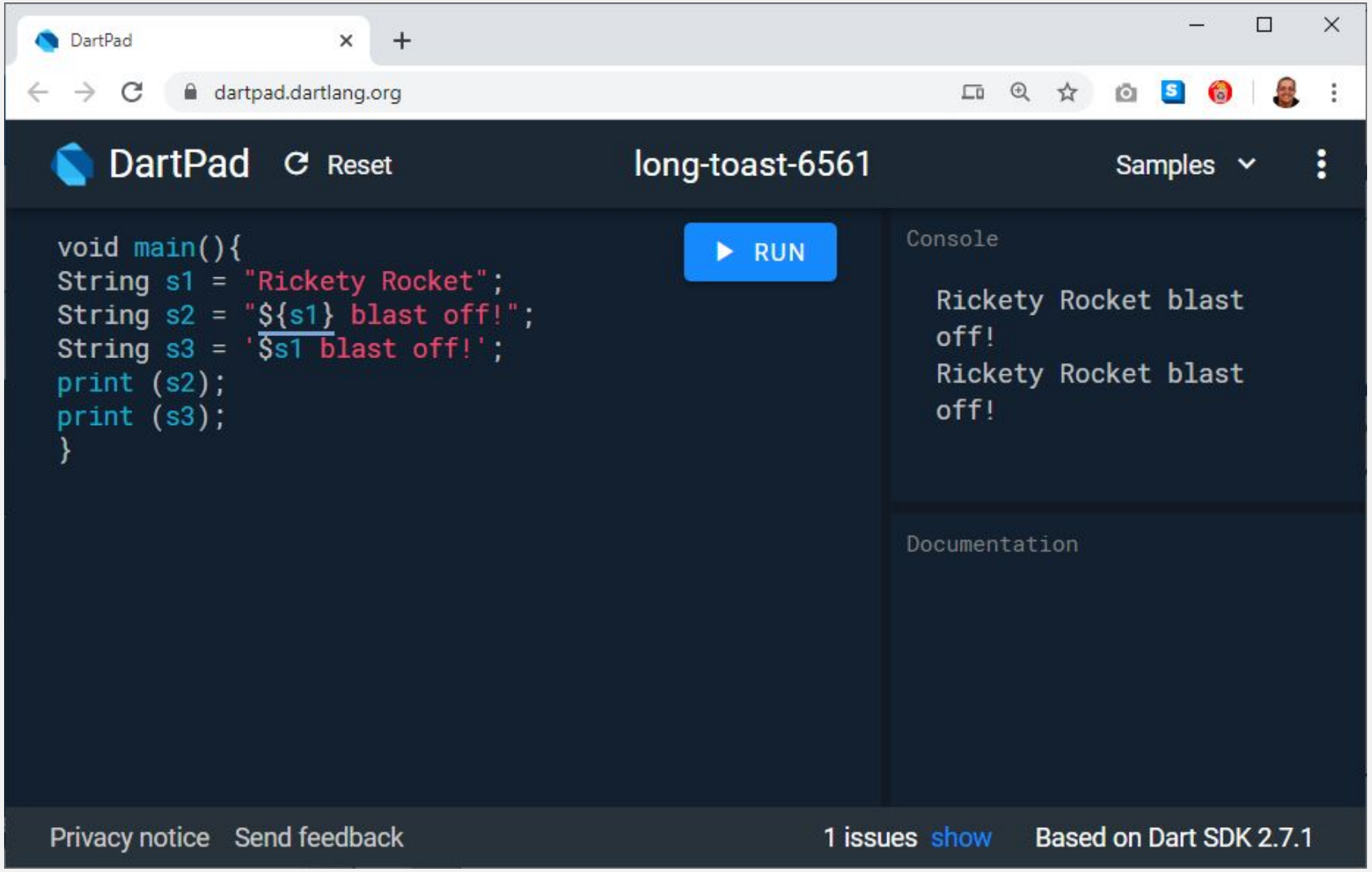
Dart

- Tipos de dados

Type	Description
int	Integers (no decimals).
double	Decimal number (double precision).
bool	Boolean true or false.
String	Immutable string.
StringBuffer	Mutable string.
RegExp	Regular expressions.
List, Map, Set	Dart provides Collection classes.
DateTime	A point in time.
Duration	A span of time.
Uri	Uniform Resource Identifier
Error	Error information

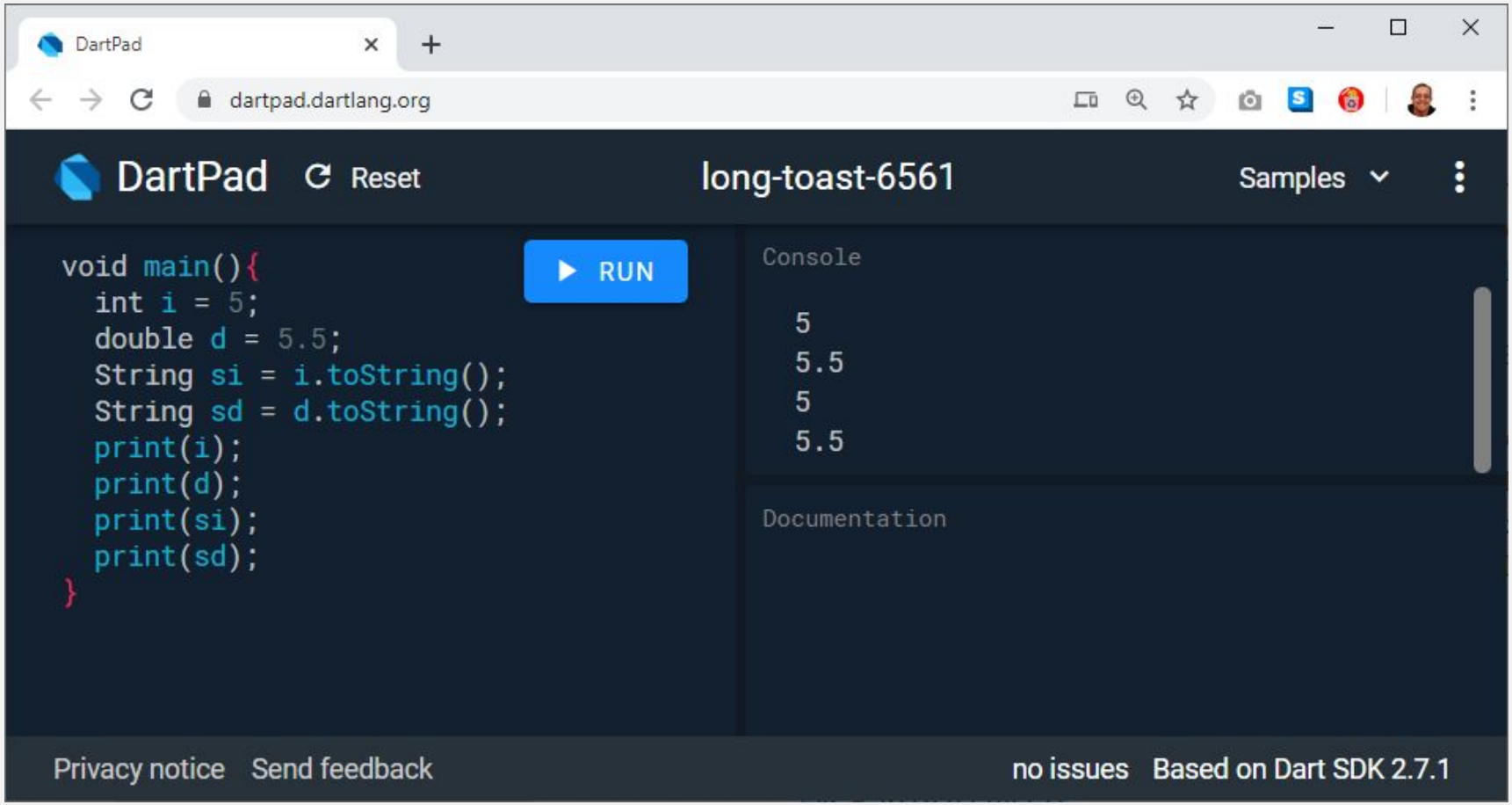
Dart

- Strings



Dart

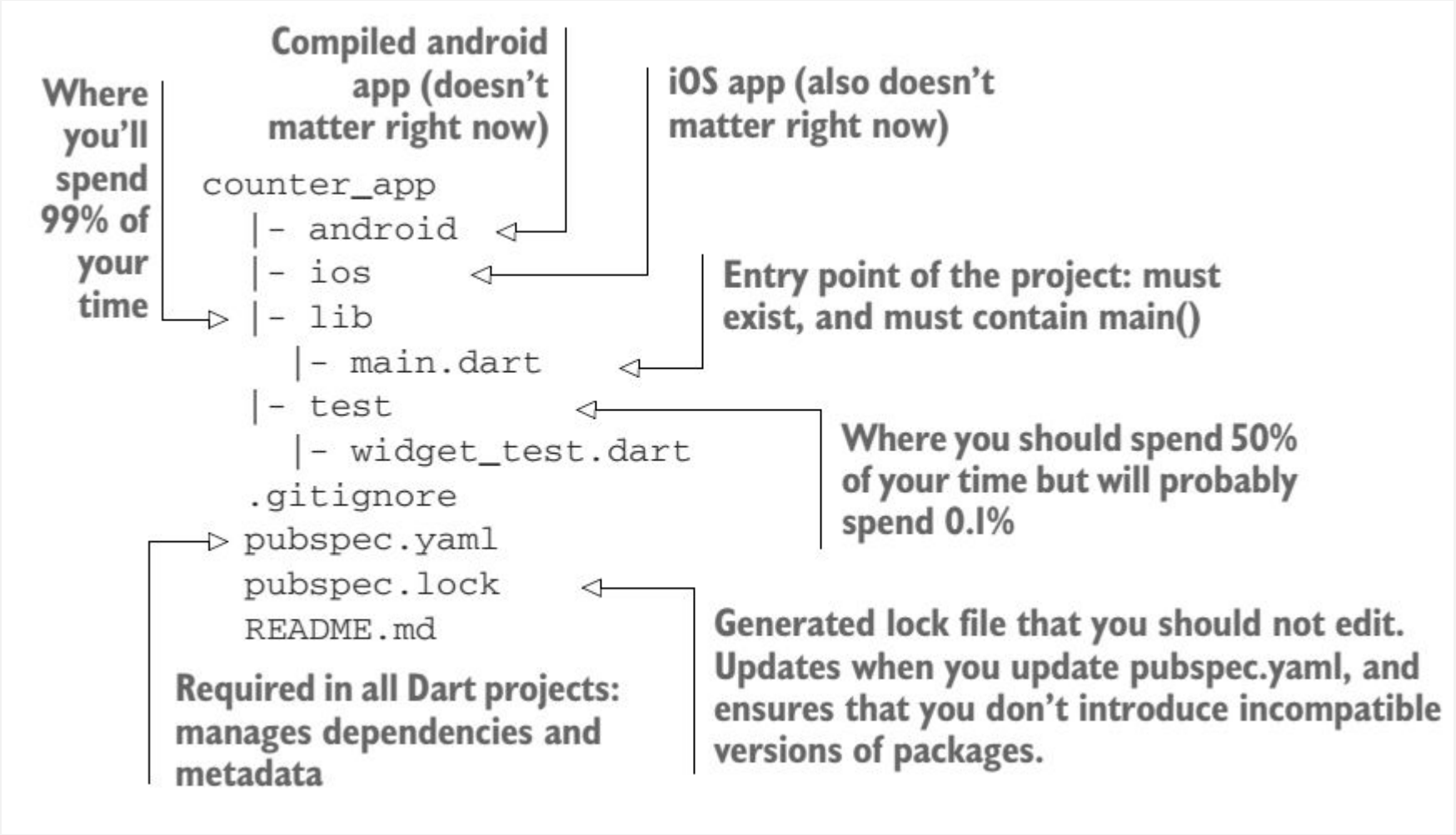
- Valores Numéricos



WIDGETS

Fundamentos

- Estrutura do projeto



Fundamentos

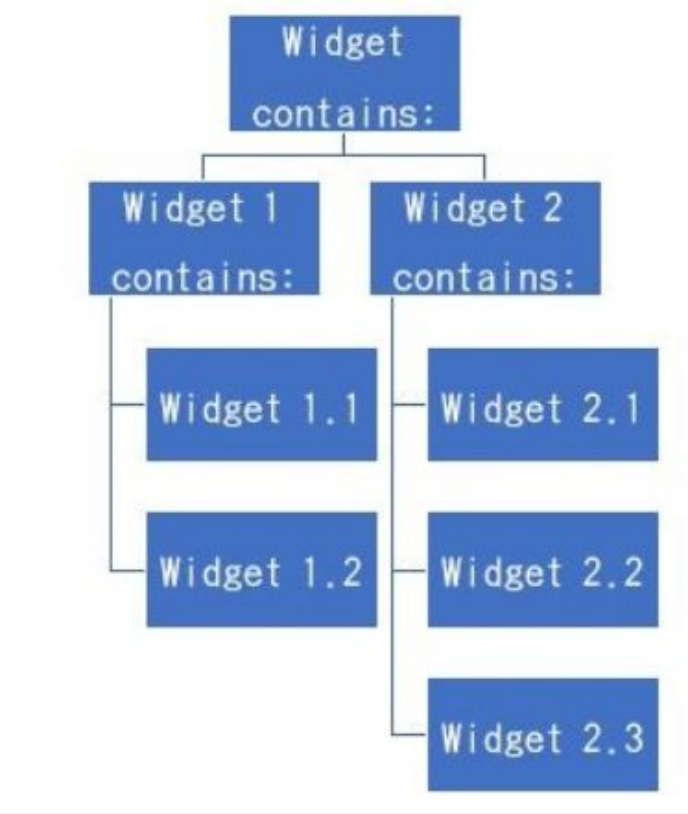
- Flutter utiliza a mesma linguagem para construção da UI e para a programação dos comportamentos

Framework	Behavior expressed in ...	UI expressed in ...
Xamarin	C#	XAML
React Native	JavaScript	JSX
NativeScript	JavaScript	XML
Flutter	Dart	Dart

Fundamentos

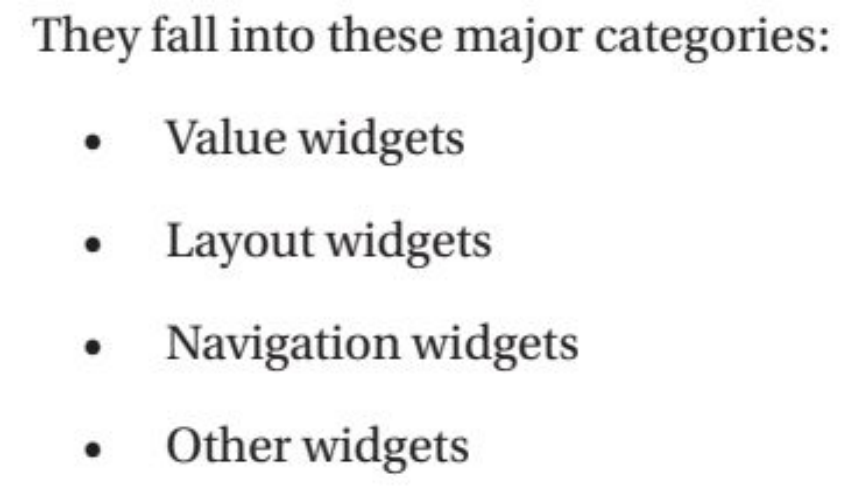
- Widgets são os blocos de construção da UI.
- A criação de uma UI é realizada por meio de uma composição de ***widgets***

Widget Tree



Fundamentos

- ***Built-in Flutter Widgets***
  - Conjunto completo de widgets.
  - As aplicações são construídas a partir da composição de widgets prontos.
  - Alguns exemplos de widgets:



https://flutter.dev/docs/development/ui/widgets

Fundamentos

- ***Value Widgets***
  - São usados exibir valores para o usuário e obter valores do usuário no App.

Checkbox	FormField	RefreshIndicator
CircularProgressIndicator	Icon	RichText
Date & Time Pickers	IconButton	Slider
DataTable	Image	Switch
DropdownButton	LinearProgressIndicator	Text
FlatButton	PopupMenuButton	TextField
FloatingActionButton	Radio	Tooltip
FlutterLogo	RaisedButton	
Form	RawImage	

Fundamentos

- ***Layout Widgets***
  - Utilizados para controlar como os widgets são apresentados na tela.

Align	FittedBox	Padding
AppBar	Flow	PageView
AspectRatio	FractionallySizedBox	Placeholder
Baseline	GridView	Row
BottomSheet	IndexedStack	Scaffold
ButtonBar	IntrinsicHeight	Scrollable
Card	IntrinsicWidth	Scrollbar
Center	LayoutBuilder	SingleChildScrollView
Column	LimitedBox	SizedBox
ConstrainedBox	ListBody	SizedOverflowBox
Container	ListTile	SliverAppBar
CustomMultiChildLayout	ListView	SnackBar
Divider	MediaQuery	Stack
Expanded	NestedScrollview	Table
ExpansionPanel	OverflowBox	Wrap

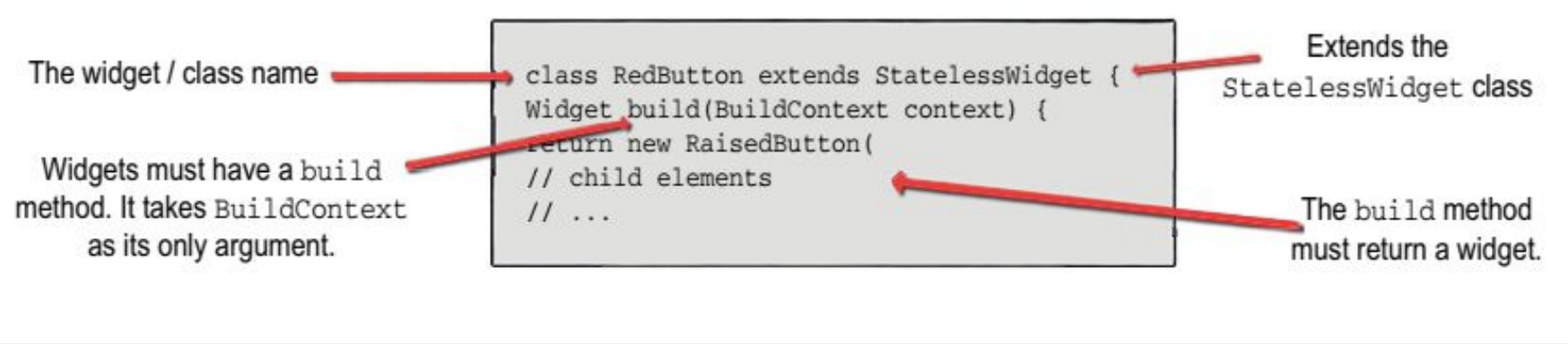
Fundamentos

- ***Navigation Widgets***
  - Quando seu aplicativo tem várias cenas ("telas", "páginas"), você precisará de alguma maneira de se mover entre eles.
  - Eles controlam como o usuário vê uma cena e depois passa para o próximo.
  - Geralmente isso é feito quando o usuário toca em um botão.

AlertDialog	MaterialApp	TabBar
BottomNavigationBar	Navigator	TabBarView
Drawer	SimpleDialog	

Fundamentos

- Todos os widgets devem conter um método denominado ***build***, o qual deve retornar um outro widget.

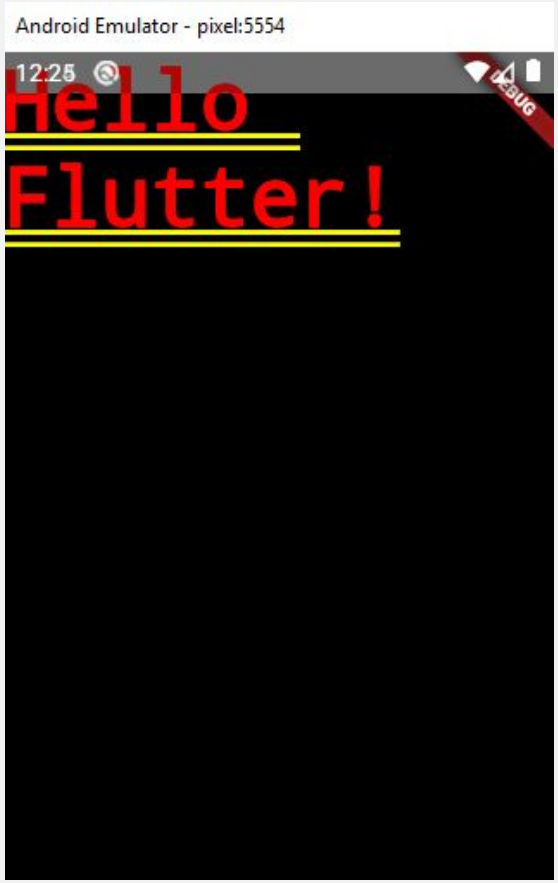


Hello Flutter

main.dart

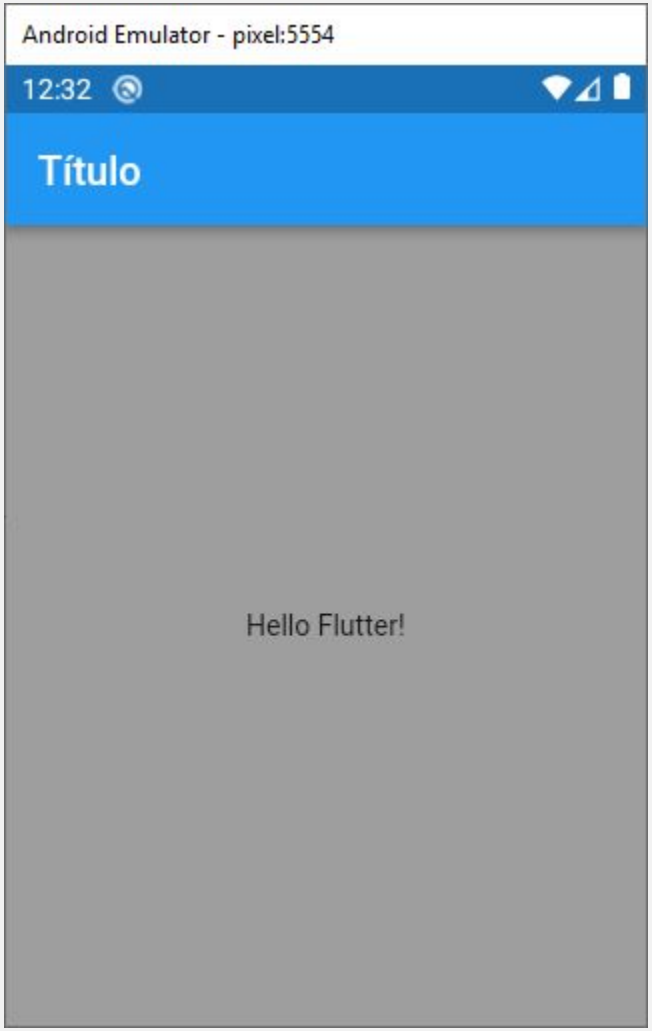
lib > main.dart > ...

```dart
import "package:flutter/material.dart";

void main() => runApp(App());

class App extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return new MaterialApp(
        title: "Meu Primeiro App",
        home: Container(
          child: Text("Hello Flutter!"),
        ));
  }
}
```



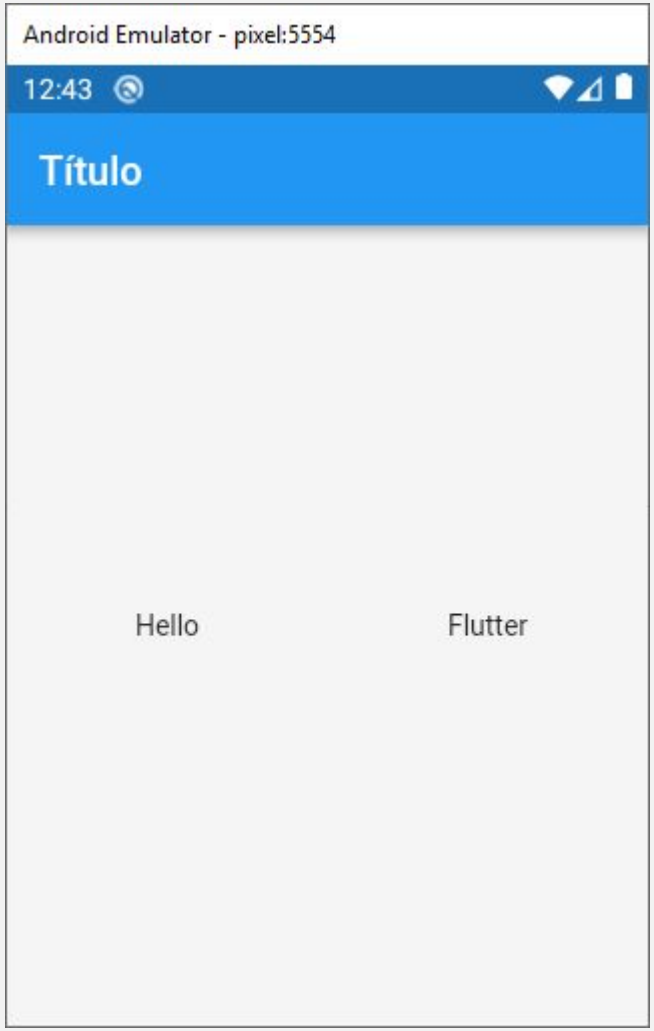
Hello Flutter

```dart
import "package:flutter/material.dart";

void main() => runApp(App());

class App extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return new MaterialApp(
        debugShowCheckedModeBanner: false,
        title: "Meu Primeiro App",
        home: Scaffold(
          appBar: AppBar(
            title: const Text('Título'),
          ),
          body: Center(
            child: Text("Hello Flutter!")
          ),
          backgroundColor: Colors.grey,
        ));
  }
}
```



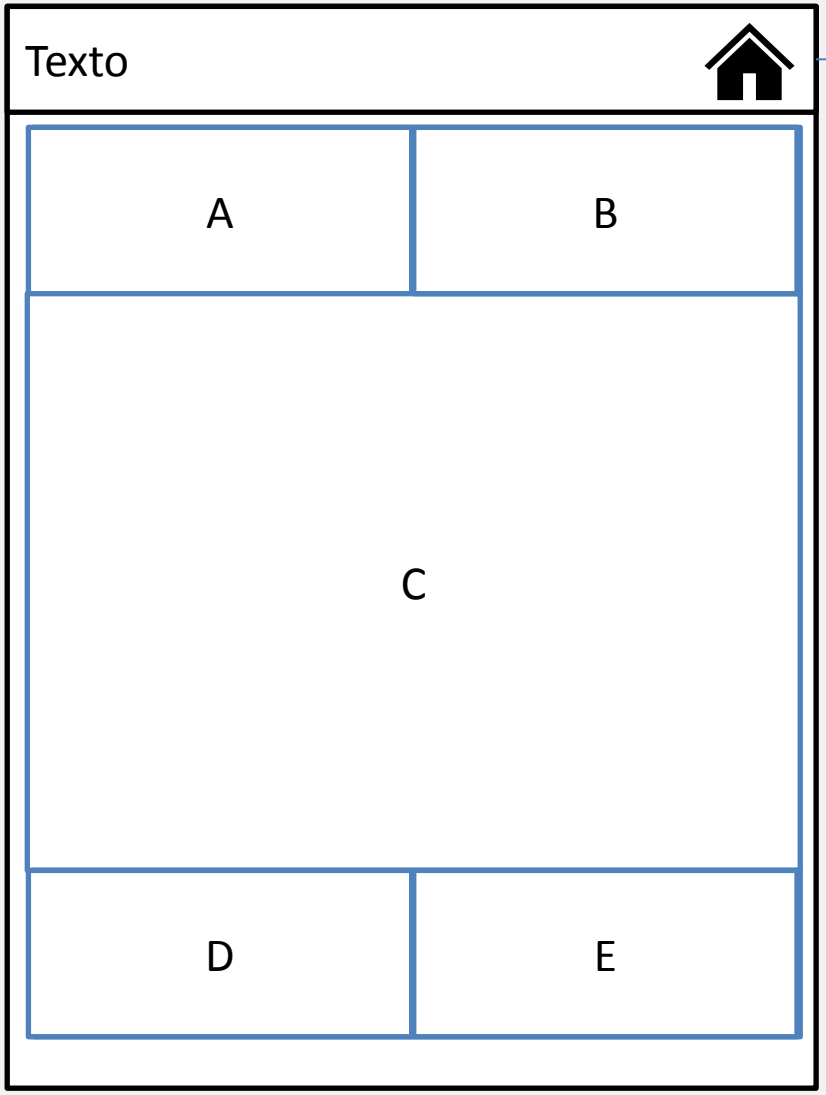
Hello Flutter

```dart
import "package:flutter/material.dart";

void main() => runApp(App());

class App extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return new MaterialApp(
        debugShowCheckedModeBanner: false,
        title: "Meu Primeiro App",
        home: Scaffold(
          appBar: AppBar(
            title: const Text('Título'),
          ), // AppBar
          body: Center(child: Row(
            children: <Widget>[
              Expanded(
                child: Text('Hello',  textAlign: TextAlign.center),
              ), // Expanded
              Expanded(
                child: Text('Flutter', textAlign: TextAlign.center),
              ), // Expanded
            ], // <Widget>[]
          ), // Row
          ), // Center
          backgroundColor: Colors.grey[100],
        )); // Scaffold // MaterialApp
  }
}
```



ATIVIDADE PRÁTICA

Elabore um App baseado no seguinte layout:



Para cada widget especificar propriedades de formatação.