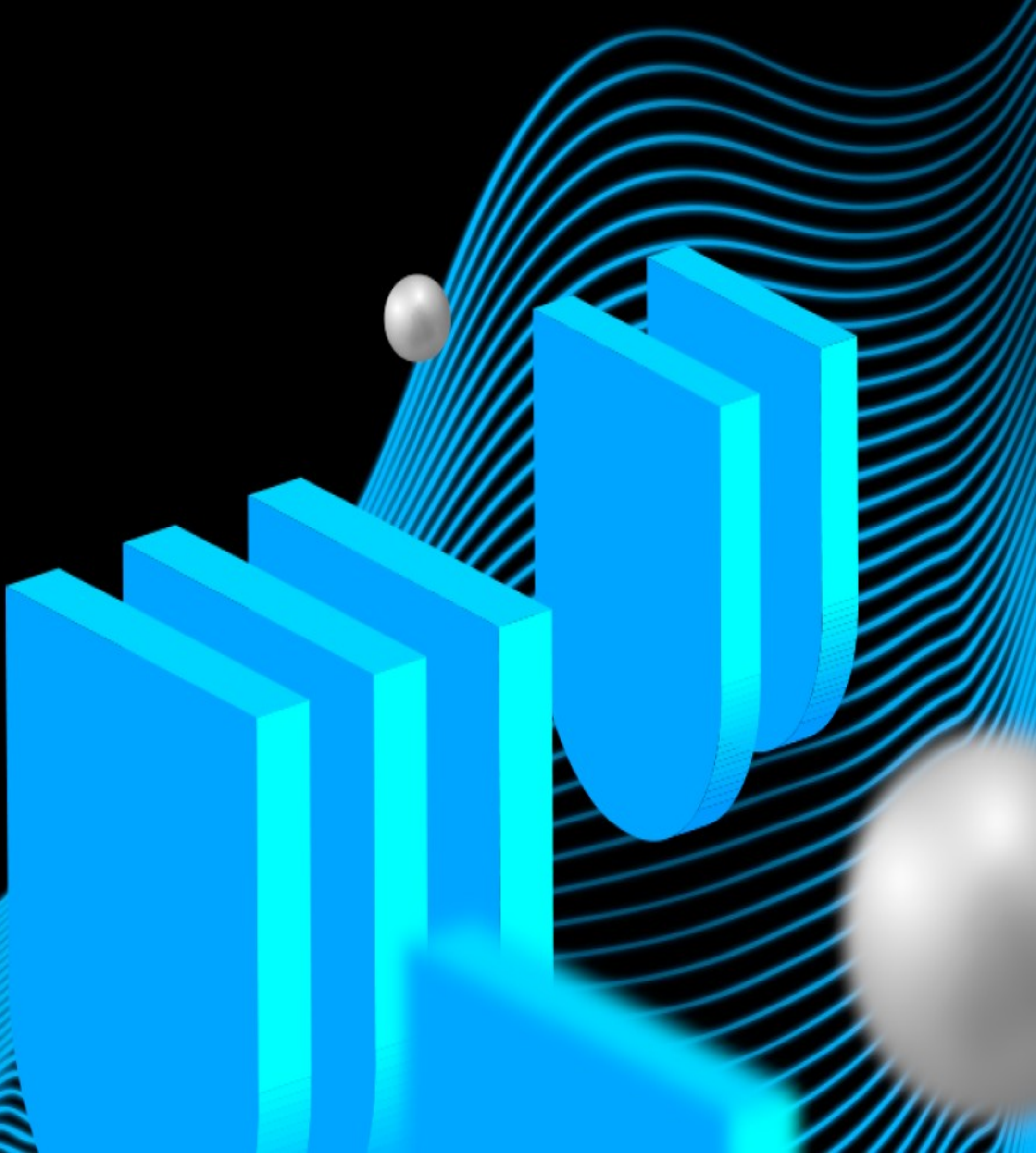
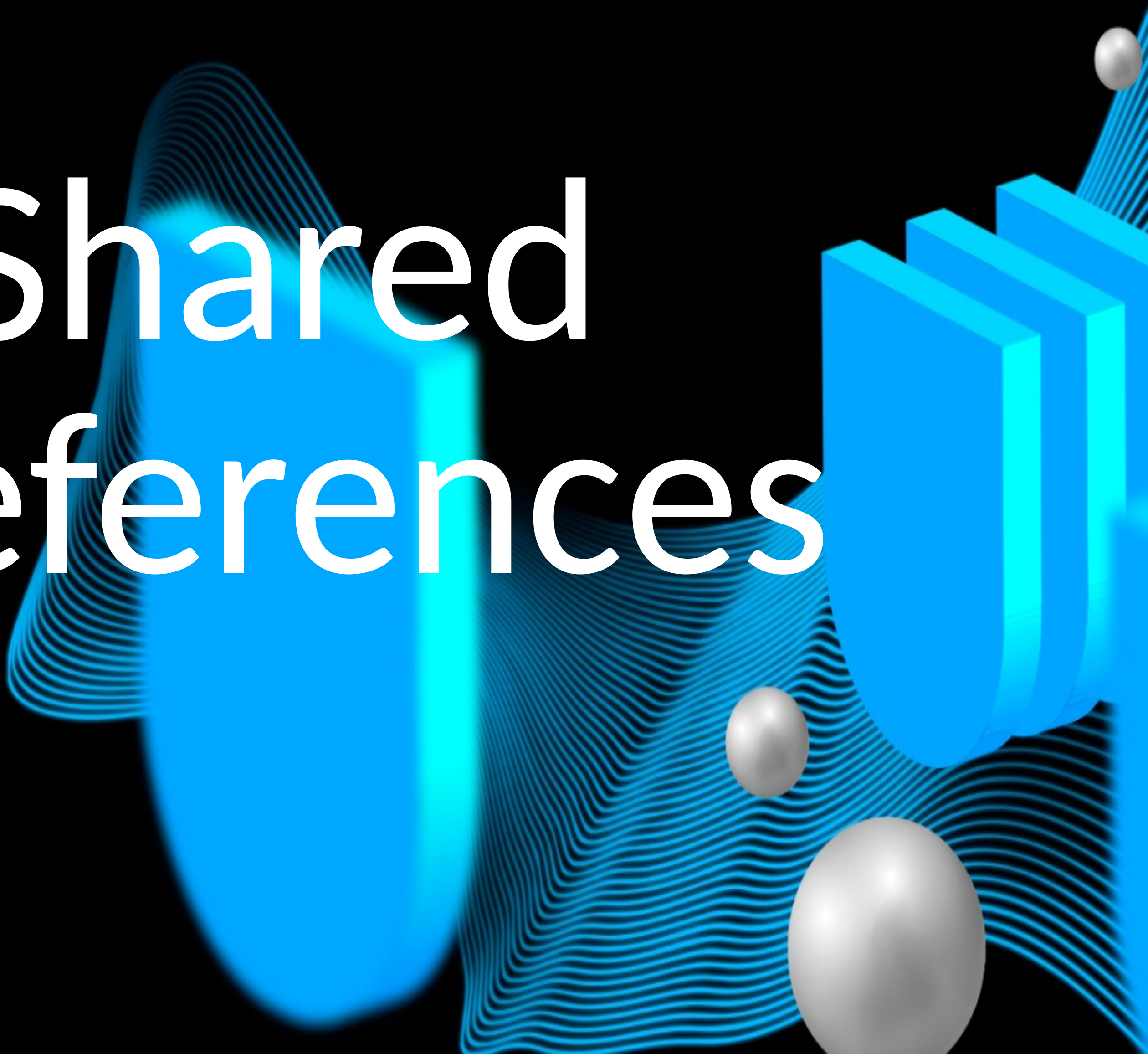
UNITAU

Universidade de Taubaté



Shared Preferences



Shared Preferences

Shared Preference é uma forma de armazenar pares chave-valor no armazenamento do dispositivo. Utiliza o NSUserDefaults no iOS e o SharedPreferences no Android. Permite o armazenamento, recuperação e remoção de valores do tipo int, double, bool, String e StringList. Pode ser usado como uma forma simples de persistência sem a necessidade de utilizar um banco de dados como SQLite. Também pode ser utilizado para armazenar a configuração da aplicação.

Shared Preferences

Crie uma nova aplicação Flutter com o nome **sharedpreferences**.

Altere o arquivo **pubspec.yaml** para incluir a dependência de **shared_preference**:

```
dependencies:
  shared_preferences: ^0.5.6+3
  flutter:
    sdk: flutter
```

Instale o pacote adicionado nas dependências.

Shared Preferences

Altere o arquivo **main.dart**:

```
import 'package:flutter/material.dart';
import 'package:shared_preferences/shared_preferences.dart';
import 'pages/home.dart';

main() async {
  WidgetsFlutterBinding.ensureInitialized();

  SharedPreferences _prefs = await
SharedPreferences.getInstance();
  if ( !_prefs.containsKey("counter") )
    _prefs.setInt('counter', 0);

  runApp(MyApp(_prefs));
}
```

Shared Preferences

```
class MyApp extends StatelessWidget {
  SharedPreferences _prefs;

  MyApp(this._prefs) : super();

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      title: 'Flutter Demo',
      theme: ThemeData(
        primarySwatch: Colors.blue,
        visualDensity:
VisualDensity.adaptivePlatformDensity,
      ),
      home: Home(prefs:_prefs),
    );
  }
}
```

Shared Preferences

Crie o subdiretório **lib/pages** e crie o arquivo **home.dart** no diretório **lib/pages**:

```
import 'package:flutter/material.dart';
import 'package:shared_preferences/shared_preferences.dart';

class Home extends StatefulWidget {
  SharedPreferences prefs;

  Home({Key key, this.prefs}) : super(key: key);

  @override
  _HomeState createState() => _HomeState();
}
```

Shared Preferences

Acrescente a classe para gerenciar o estado da tela no arquivo home.dart:

```
class _HomeState extends State<Home> {

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text("Shared Preferences"),
      ),
```

Shared Preferences

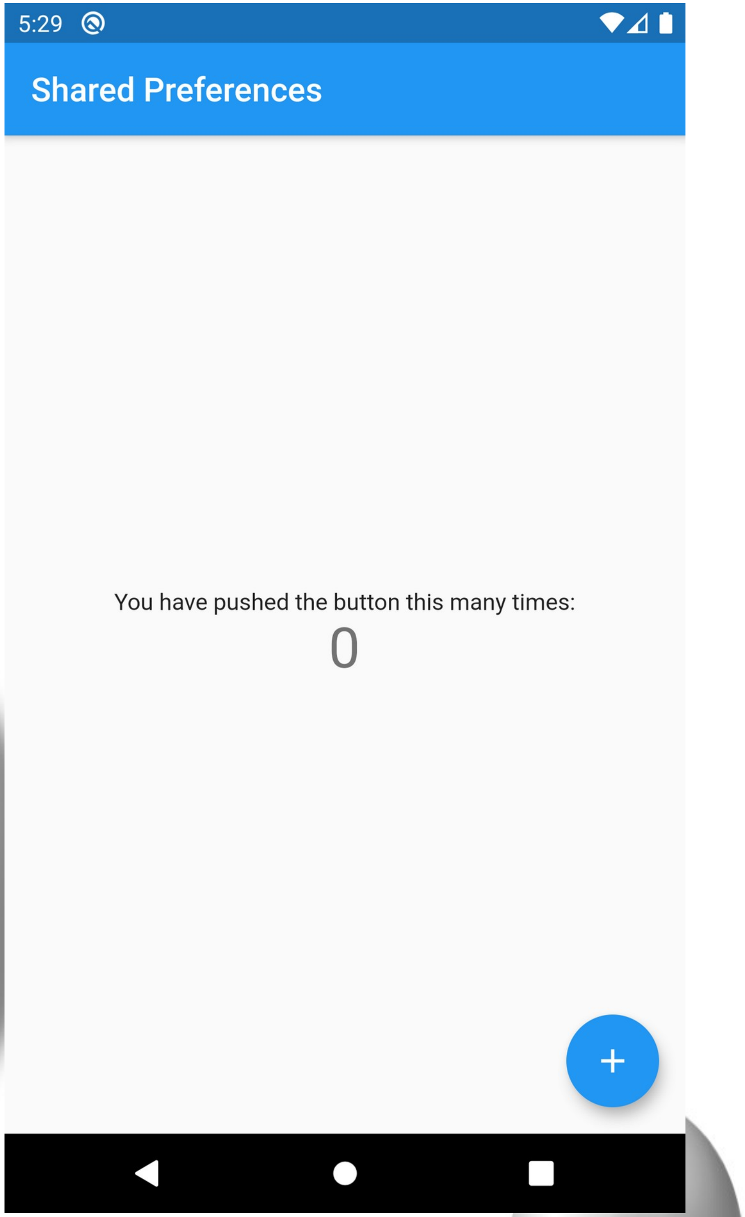
```
      body: Center(
        child: Column(
          mainAxisAlignment: MainAxisAlignment.center,
          children: <Widget>[
            Text(
              'You have pushed the button this many times:',
            ),
            Text(
              '${widget.prefs.getInt("counter")}',
              style: Theme.of(context).textTheme.display1,
            ),
          ],
        ),
      ),
      floatingActionButton: FloatingActionButton(
        onPressed: _incrementCounter,
        tooltip: 'Increment',
        child: Icon(Icons.add),
      ),
    );
  }
}
```

Shared Preferences

Acrescente o método para incrementar o contado na classe _HomeState:

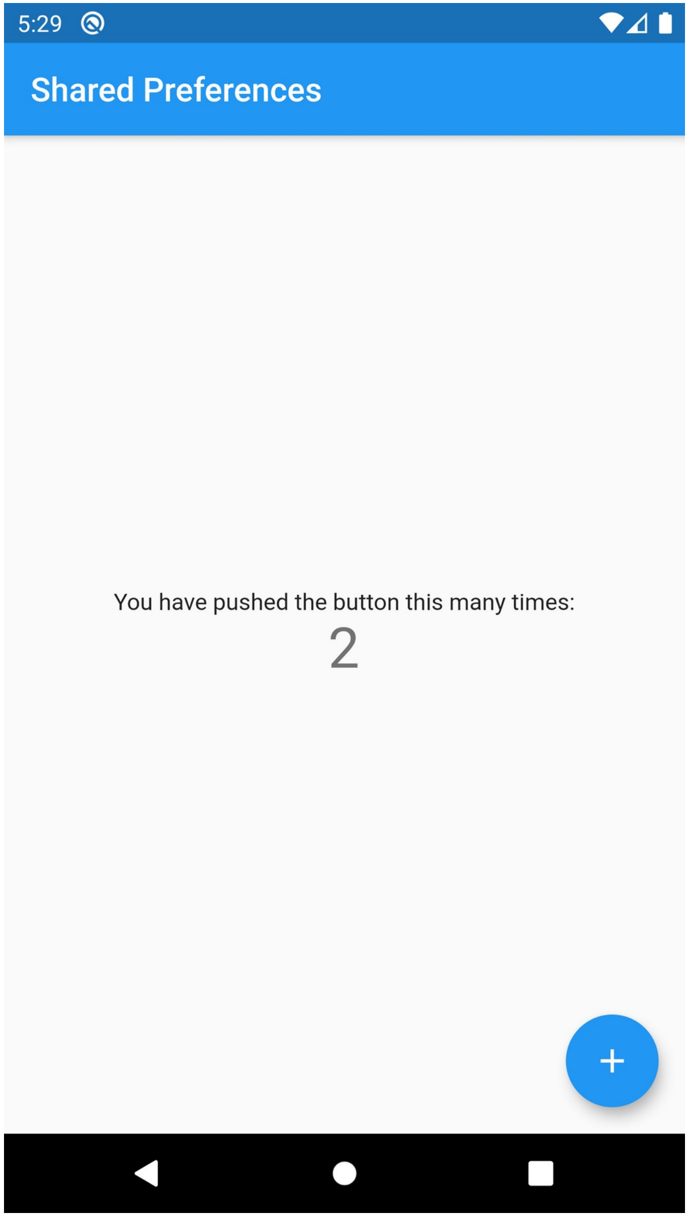
```
  void _incrementCounter() {
    setState(() {
      widget.prefs.setInt('counter', 
widget.prefs.getInt("counter")+1);
    });
  }
```

Shared Preferences



You have pushed the button this many times:

0



You have pushed the button this many times:

2

Preferences

O pacote preferences provê widgets para facilitar a criação de uma tela para edição das shared preferences.

Crie uma nova aplicação Flutter com o nome preferences_app.

Altere o arquivo pubspec.yaml para incluir as dependências:

```
dependencies:
  preferences: ^5.1.0
  dynamic_theme:
  flutter:
    sdk: flutter
```

Instale os pacotes adicionados nas dependências.

Preferences

Altere o arquivo **main.dart**:

```
import 'package:flutter/material.dart';
import 'package:preferences/preferences.dart';
import 'package:dynamic_theme/dynamic_theme.dart';
import 'pages/home.dart';
import 'utils/theme.dart';

void main() async {
  WidgetsFlutterBinding.ensureInitialized();

  await PrefService.init(prefix: 'pref_');
  PrefService.setDefaultValues({'brightness': 'light'});
  PrefService.setDefaultValues({'color': 'Blue'});
  PrefService.setDefaultValues({'title': 'Preferences'});
  PrefService.setDefaultValues({'show_label': true});
  PrefService.setDefaultValues({'show_counter': true});
  PrefService.setDefaultValues({'action_increment': true});
  PrefService.setDefaultValues({'action_decrement': true});
  PrefService.setDefaultValues({'counter': 0});
  runApp(MyApp());
}
```

Preferences

```
class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {

    return DynamicTheme(
      defaultBrightness: Brightness.light,
      data: (brightness) => getTheme(),
      themedWidgetBuilder: (context, theme) {
        return MaterialApp(
          debugShowCheckedModeBanner: false,
          title: 'Preferences Demo',
          theme: theme,
          home: Home(),
        );
      }
    );
  }
}
```

Preferences

Crie os subdiretórios lib/utils e crie o arquivo theme.dart no diretório lib/utils:

```
import 'package:flutter/material.dart';
import 'package:preferences/preferences.dart';

ThemeData getTheme() {
  String _brightnessValue =
PrefService.getString('brightness');
  Brightness _brightness = Brightness.light;
  if ( _brightnessValue == "dark")
    _brightness = Brightness.dark;
  String _colorValue = PrefService.getString('color');
  Color _color = Colors.blue;
  if ( _colorValue == "Green")
    _color = Colors.green;
  else if ( _colorValue == "Red")
    _color = Colors.red;
  return ThemeData(
    brightness: _brightness,
    primaryColor: _color, );
}
```

Preferences

Crie os subdiretórios lib/pages e crie o arquivo home.dart no diretório lib/pages:

```
import 'package:flutter/material.dart';
import 'package:preferences/preferences.dart';
import 'configuration.dart';

class Home extends StatefulWidget {

  @override
  _HomeState createState() => _HomeState();
}
```

Preferences

Acrescente a classe para gerenciar o estado da tela no arquivo **home.dart**:

```
class _HomeState extends State<Home> {

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text(PrefService.getString('title')),
        actions: <Widget>[
          IconButton(icon: Icon(Icons.settings),
            onPressed: () async {
              await Navigator.push(context,
                MaterialPageRoute(builder: (context) =>
Configuration()),
              );
              setState(() {});
            },
          ),
        ],
      ),
```

Preferences

```
      body: Center(
        child: Column(
          mainAxisAlignment: MainAxisAlignment.center,
          children: <Widget>[
            Text(PrefService.getBool('show_label')?
              'You have pushed the button this many times:'
              :"No Label",
            ),
            Text(PrefService.getBool('show_counter')?
              '${PrefService.getInt('counter')}':"No
Counter",
              style: Theme.of(context).textTheme.display1,
            ),
          ],
        ),
      ),
```

Preferences

```
      floatingActionButton: Row(
        mainAxisAlignment: MainAxisAlignment.spaceEvenly,
        children: <Widget>[
          FloatingActionButton(
            onPressed:
PrefService.getBool('action_decrement')?
              _decrementCounter:null,
            tooltip: 'Decrement',
            child: Icon(Icons.remove),
            backgroundColor:
PrefService.getBool('action_decrement')?
              Colors.tealAccent:Colors.grey,
            heroTag: null
          ),
```

Preferences

```
          FloatingActionButton(
            onPressed:
PrefService.getBool('action_increment')?
              _incrementCounter:null,
            tooltip: 'Increment',
            child: Icon(Icons.add),
            backgroundColor:
PrefService.getBool('action_increment')?
              Colors.tealAccent:Colors.grey,
            heroTag: null
          ),
        ],
      ),
    );
  }
}
```

Preferences

Acrescente o método para incrementar o contado na classe _HomeState:

```
  void _incrementCounter() {
    setState(() {
      PrefService.setInt('counter',
PrefService.getInt('counter')+1);
    });
  }
```

Acrescente o método para decrementar o contado na classe _HomeState:

```
  void _decrementCounter() {
    setState(() {
      PrefService.setInt('counter',
PrefService.getInt('counter')-1);
    });
  }
```

Preferences

Crie o arquivo **configuration.dart** no diretório **lib/pages**:

```
import 'package:flutter/material.dart';
import 'package:preferences/preferences.dart';
import 'package:dynamic_theme/dynamic_theme.dart';
import '../utils/theme.dart';

class Configuration extends StatefulWidget {

  @override
  _ConfigurationState createState() =>
_ConfigurationState();
}
```

Preferences

Acrescente a classe para gerenciar o estado da tela no arquivo configuration.dart:

```
class _ConfigurationState extends State<Configuration> {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text("Configuracao"),
      ),
      body: PreferencePage([
        PreferenceTitle('Theme'),
```

Preferences

```
        PreferenceDialogLink(
          'Brightness',
          dialog: PreferenceDialog(
            [
              RadioPreference('Light', 'light', 
'brightness',
                onSelect: () => 
DynamicTheme.of(context).setThemeData(getTheme()),
              ),
              RadioPreference('Dark', 'dark', 'brightness',
                onSelect: () => 
DynamicTheme.of(context).setThemeData(getTheme()),
              ),
            ],
            title: 'Select an option',
            cancelText: 'Close',
          ),
        ),
```

Preferences

```
        DropdownPreference('Color',
          'color',
          defaultVal: 'Blue',
          values: ['Blue', 'Green', 'Red'],
          onChange: (_value) =>
DynamicTheme.of(context).setThemeData(getTheme()),
        ),

        PreferenceTitle('Content'),

        PreferenceDialogLink('Show',
          dialog: PreferenceDialog(
            [
              CheckboxPreference('Label', 'show_label'),
              CheckboxPreference('Counter', 'show_counter'),
            ],
            title: 'Enabled Content',
            cancelText: 'Cancel',
            submitText: 'Save',
            onlySaveOnSubmit: true,
          ),
        ),
```

Preferences

```
PreferencePageLink(
  'Actions',
  leading: Icon(Icons.code),
  trailing: Icon(Icons.keyboard_arrow_right),
  page: PreferencePage([
    PreferenceTitle('Enable'),
    SwitchPreference(
      'Increment',
      'action_increment',
      defaultVal: true,
      desc: 'Enable incremment buttom',
    ),
    SwitchPreference(
      'Decrement',
      'action_decrement',
      defaultVal: true,
      desc: 'Enable decremment buttom',
    ),
  ]),
),
```

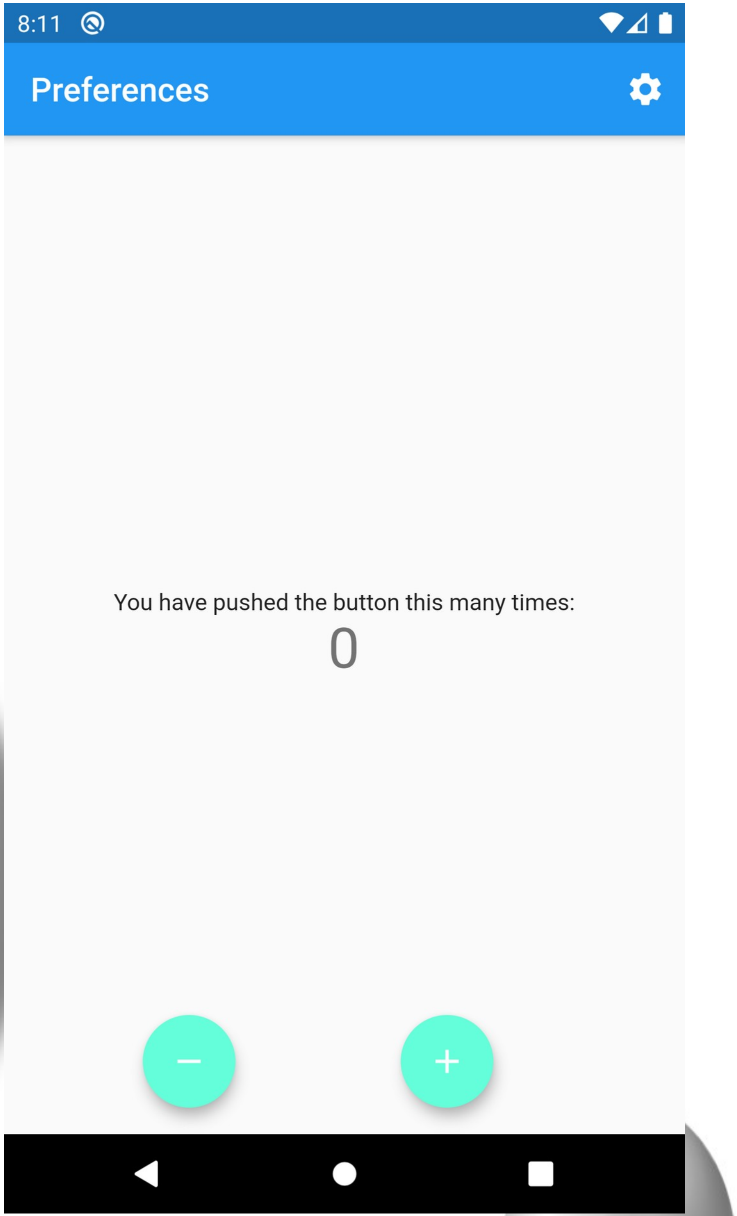
Preferences

```
PreferenceTitle('Advanced'),

CheckboxPreference(
  'Enable Advanced Features',
  'advanced_enabled',
  onChange: () => setState(() {}),
),

PreferenceHider([
  TextFieldPreference(
    'Title',
    'title',
  ),
], '!advanced_enabled'),
```

Preferences

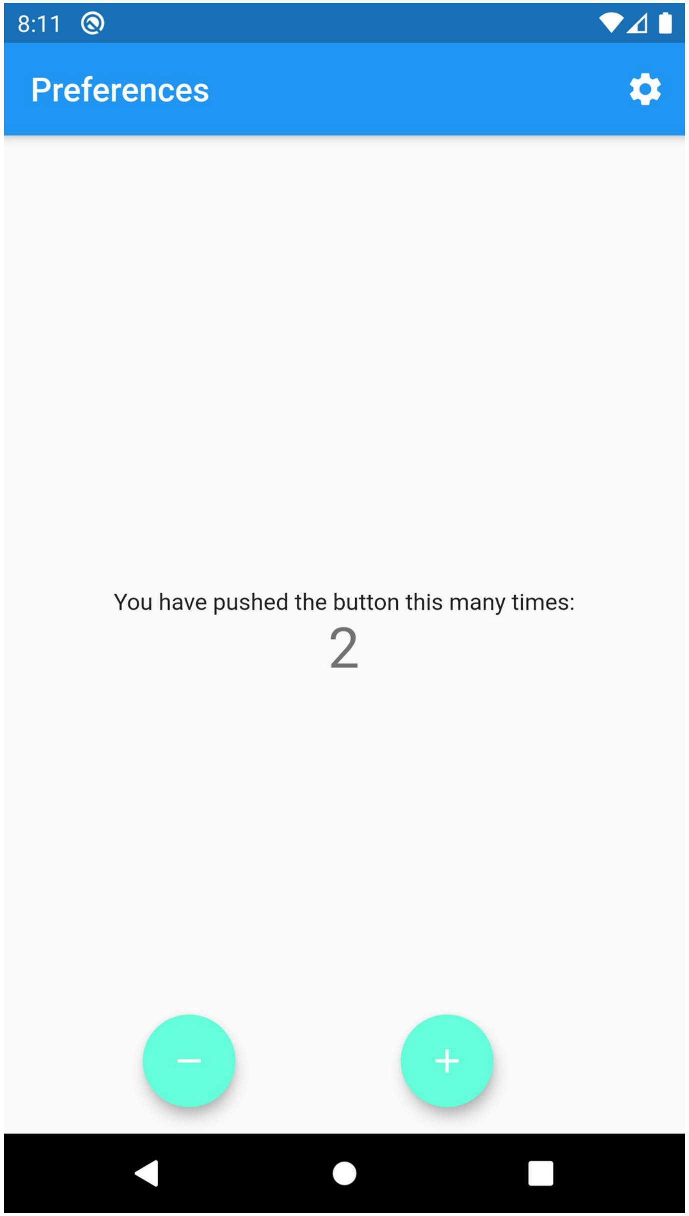
```
        PreferenceTitle('Display'),

        PreferenceText(
          "Counter: ${PrefService.getInt('counter')}",
          style: TextStyle(color: Colors.grey),
        ),
      ]),
    );
  }
}
```

Preferences



You have pushed the button this many times:

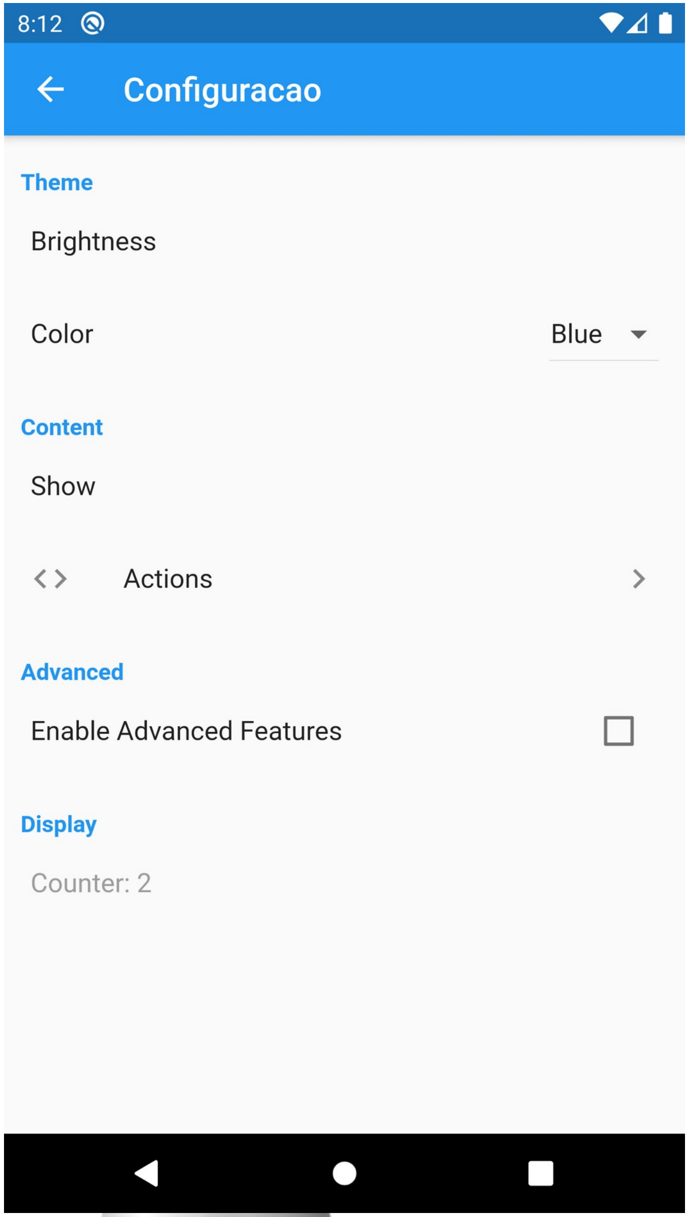
0



You have pushed the button this many times:

2

Preferences



Configuracao

Theme

Brightness

Color Blue

Content

Show

Actions

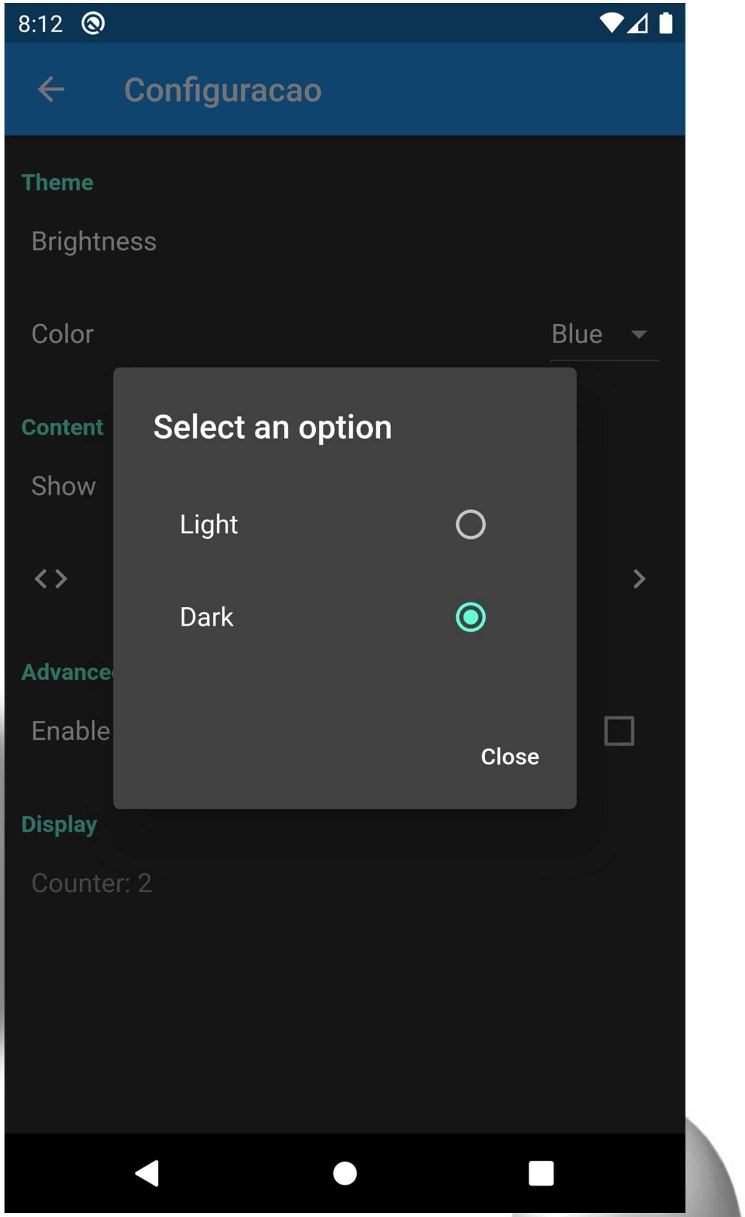
Advanced

Enable Advanced Features □

Display

Counter: 2

Preferences



Theme

Brightness

Color Blue

Content

Show

Advance

Enable □

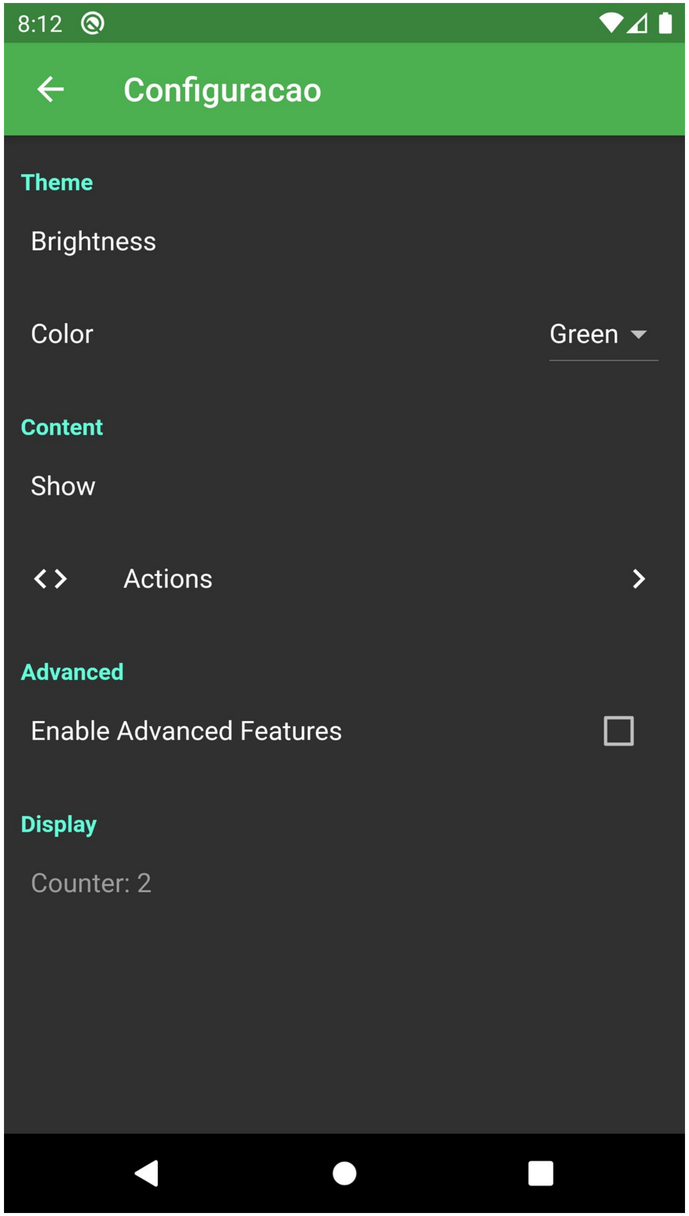
Display

Counter: 2

Select an option

Light ○

Dark ●



Theme

Brightness

Color Green

Content

Show

Actions

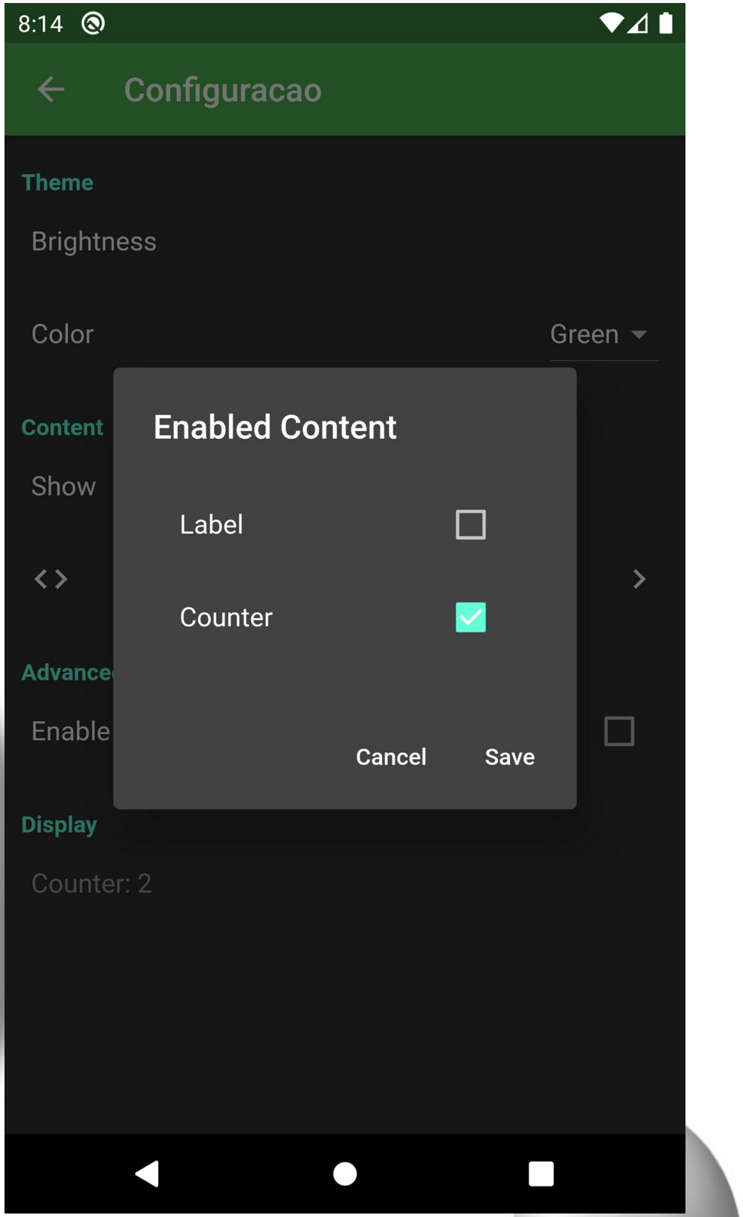
Advanced

Enable Advanced Features □

Display

Counter: 2

Preferences



Theme

Brightness

Color Green

Content

Show

Advance

Enable □

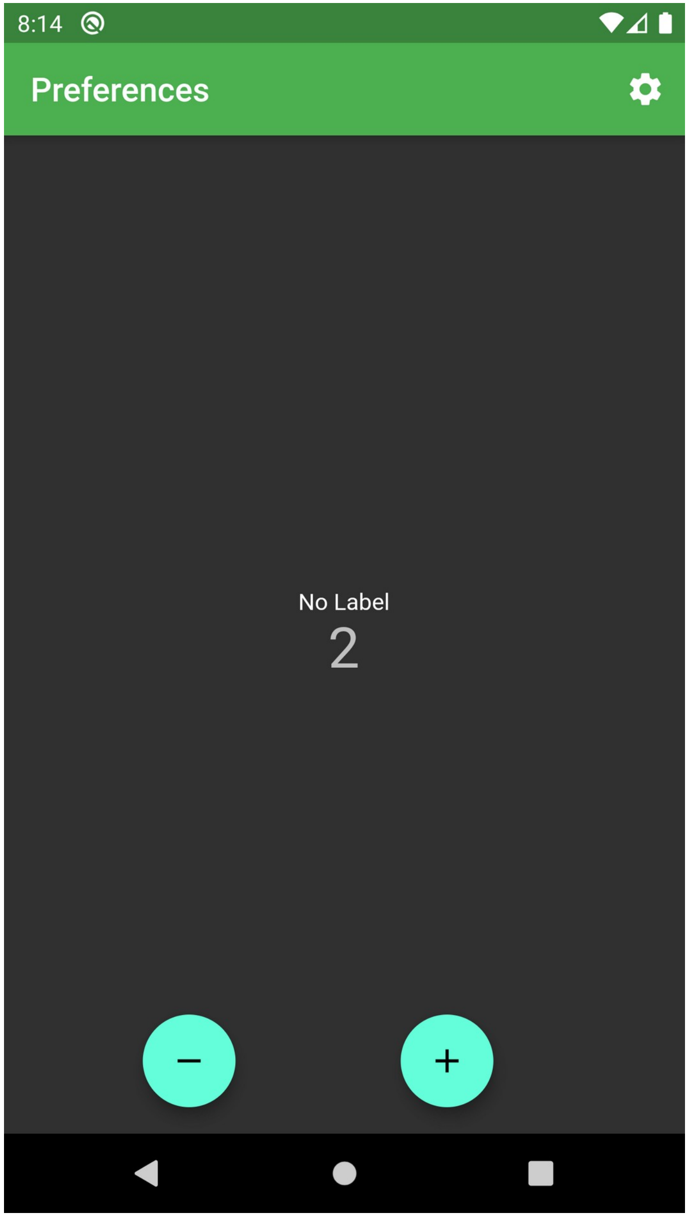
Display

Counter: 2

Enabled Content

Label □

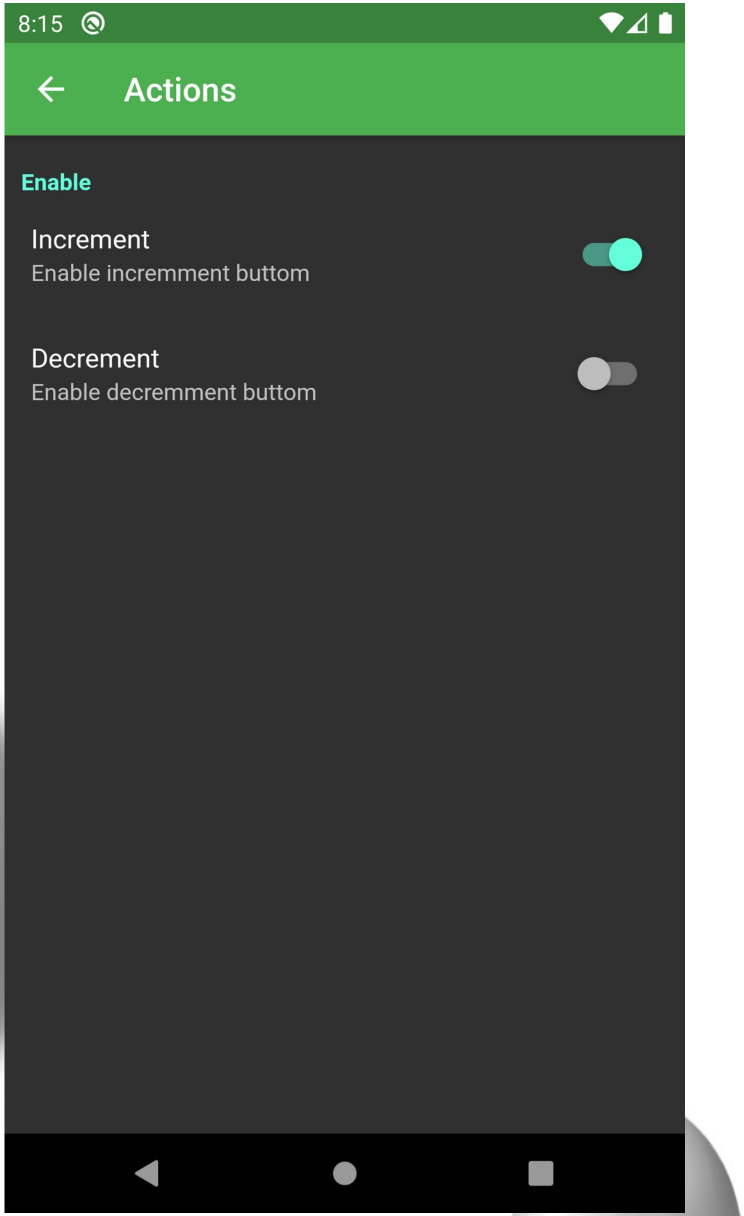
Counter ☑



No Label

2

Preferences



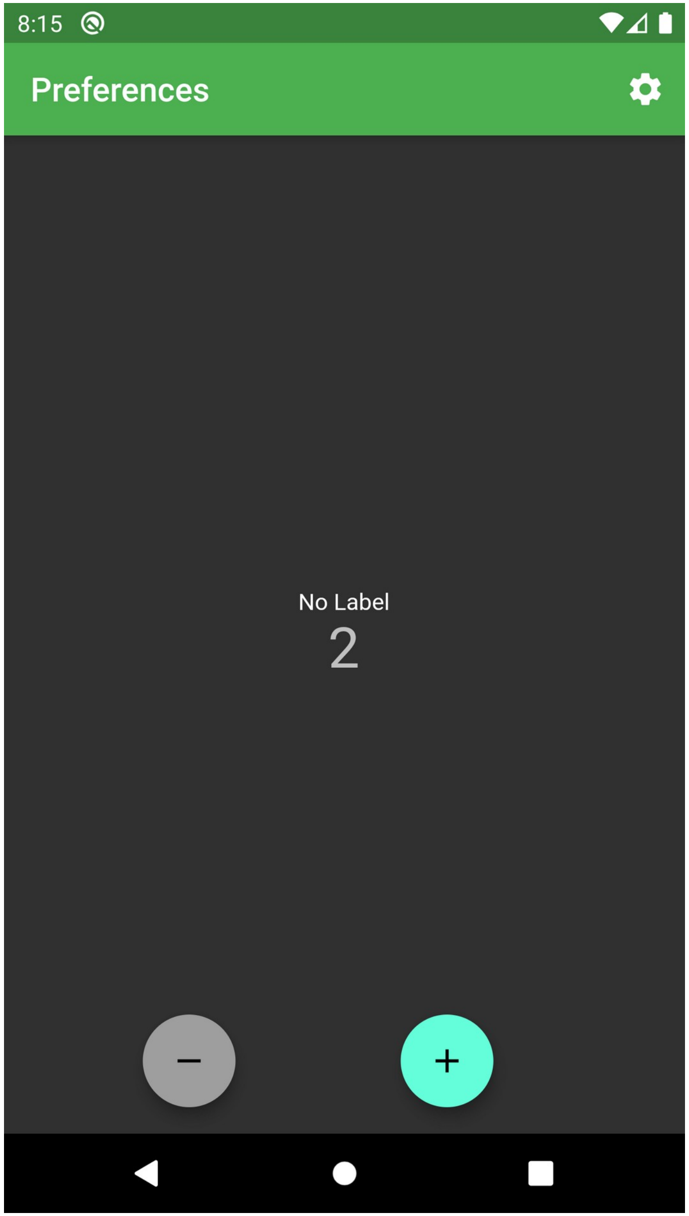
8:15

Actions

Enable

Increment
Enable incremment buttom

Decrement
Enable decremment buttom

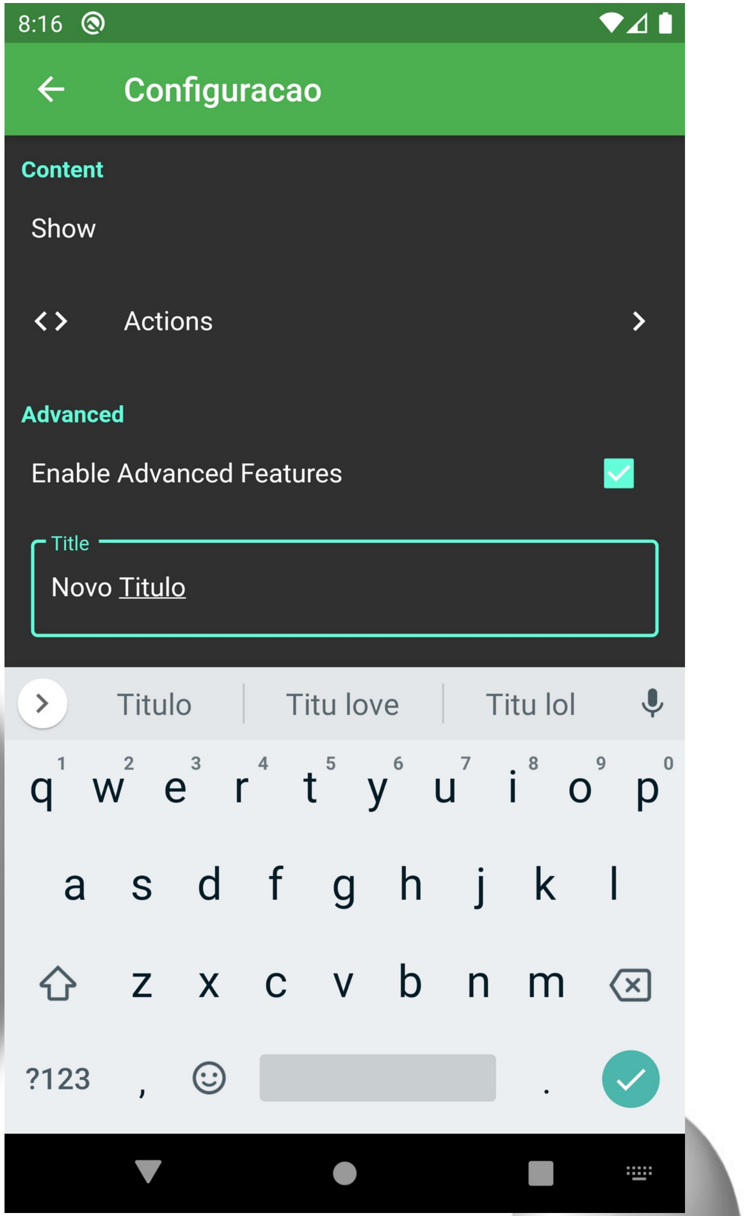


8:15

Preferences

No Label
2

Preferences



8:16

Configuracao

Content

Show

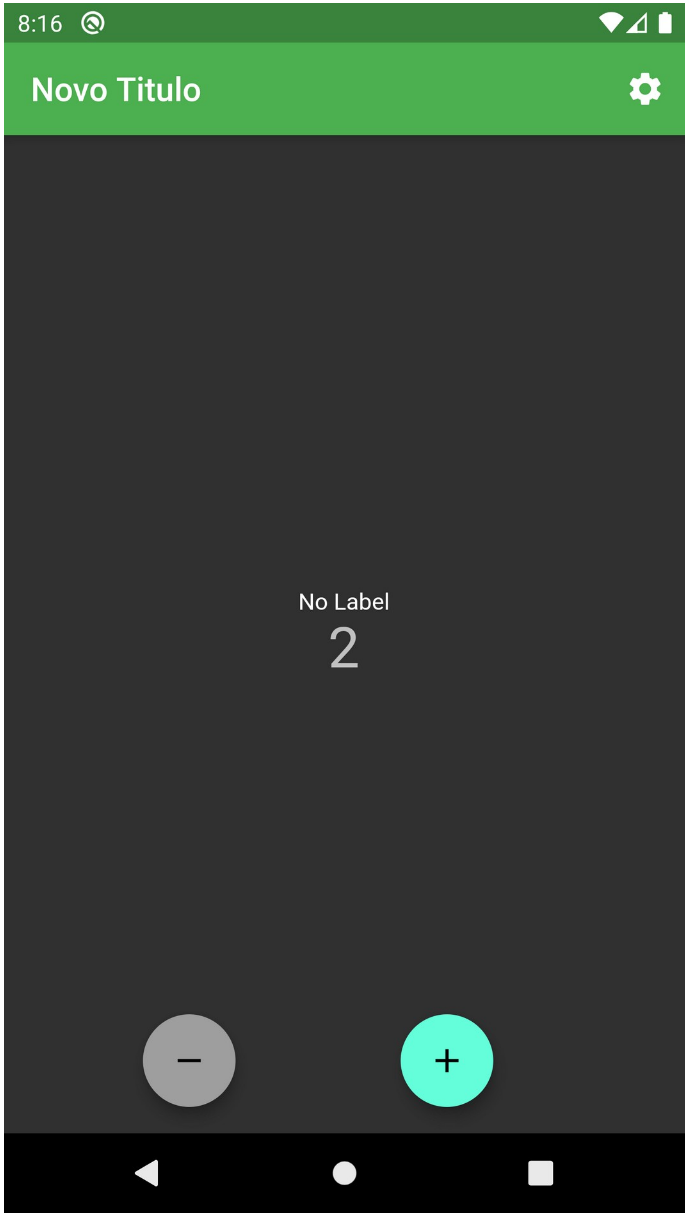
Actions

Advanced

Enable Advanced Features ☑

Title

Novo Titulo



8:16

Novo Titulo

No Label

2