

DCC917A – TÓPICOS ESPECIAIS III: DESENVOLVIMENTO DE APLICATIVOS MÓVEIS

AULA 01

Carlos Bruno Oliveira Lopes

*Engenheiro de Computação
Mestre em Ciência da Computação*

Plano de Ensino

- Objetivo
 - Capacitar o aluno no desenvolvimento de aplicativos móveis usando as tecnologias mais recentes e populares do mercado.

Plano de Ensino

EMENTA RESUMIDA

- Flutter;
- Criação de projetos;
- Criação de Apps;
- Programação na Linguagem Dart;
- Componentes de interface;
- Componentes de entrada de dados;
- Navegação;
- Consumo de serviços web (webservice);
- Manipulação de dados;
- Mídias;

Plano de Ensino

Biblioteca Básica

ZAMMETTI, Frank. **Flutter na Prática: Melhore seu Desenvolvimento Mobile com o SDK**. Novatec, 2020.

BURD, Barry. **Flutter for Dummies**. For Dummies, 2020.

BIESSEK, Alessandro. **Flutter for Beginners: An introductory guide to building cross-platform mobile applications with Flutter and Dart 2**. Packt, 2019.

Plano de Ensino

Biblioteca Complementar:

SANDE, Jonathan; GALLOWAY, Matt. **Dart Apprentice (First Edition): Beginning Programming with Dart**. R.R. Bowker. 2021

Desenvolvimento de aplicativos móveis

- Criar aplicativos móveis com aparência e funcionamento de aplicativos nativos e que também sejam multiplataforma é uma proposta complexa!
- Escrever um código nativo
 - Fornecem desempenho e recursos nativos para o aplicativo;
 - Problema:
 - Escrever o mesmo código diversas vezes para diferentes plataforma;
 - » O cliente não gosta de pagar por isso;
 - Solução:
 - Escrever o código em um plataforma crossover;
 - Ele faz uso de uma outra linguagem como base código para diversas plataformas (ex.: HTML)
 - Desvantagem:
 - » Em geral, recursos nativos não estão disponíveis e há perda de desempenho;

Desenvolvimento de aplicativos móveis

Flutter

- É uma plataforma que fornece um meio de escrevermos uma única base de código que funcione igualmente bem no Android e no iOS e distribua ao mesmo tempo desempenho e recursos nativos.
- É uma criação do Google.
 - Ela surgiu com nome “Sky”, em 2015, na Dart Developer Summit;
 - Inicialmente, só executava para o Android, mas depois de um tempo foi portado para iOS da Apple;
 - Em 4 de dezembro de 2018 foi lançado o Flutter 1.0, a primeira versão “estável”;
 - Principal característica inicial era que ele conseguia renderizar Uis de aplicativos com 120 quadros por segundo (Poucas multiplataforma consegue fazer isso);
 - **Diferencial:** Flutter renderiza os próprios componentes de UI.

Desenvolvimento de aplicativos móveis

Flutter

- Consegue renderizar os próprios componentes de UI;
 - Ou seja, a plataforma não solicita ao SO subjacente que o renderize.
- Devido a isso ele consegue fornece widgets de design específicos;
 - Ele oferece dois conjuntos de widgets:
 - Material Design – design padrão do Android
(<https://api.flutter.dev/flutter/material/material-library.html>)
 - Cupertino Design – design padrão do Apple
(<https://api.flutter.dev/flutter/cupertino/cupertino-library.html>)

Desenvolvimento de aplicativos móveis

Flutter

– Características:

- Usa plataforma Dart (linguagem);
- Engine Gráfico Skie (biblioteca open source e compacta, escrita em C++);
- Interface sobre os SDKs nativos das duas plataformas, Android e iOS, chamada biblioteca foundation;
- Widgets

Desenvolvimento de aplicativos móveis

Dart (www.dartlang.org)

- É uma linguagem de programação de uso geral;
 - Ela é empregada na construção de todo tipo de coisa, de aplicativos web a códigos de servidor e aplicativos IoT e assim por diante;
- Foi criado pela Google em meados de 2011, sendo apresentada na conferência GOTO em Aarhus, Dinamarca;
 - Sua versão inicial, 1.0, saiu em novembro de 2013, cerca de dois anos antes do Flutter ser lançado;

Desenvolvimento de aplicativos móveis

Dart

– Características:

- É orientado a objetos;
- É uma linguagem com coleta de lixo (garbage-collected), logo, não precisa-se preocupar com alocação/desalocação de memória;
- Tem um estilo de sintaxe baseado em C;
- Suporta recursos de linguagem comuns como interfaces, classes abstratas, genéricos reificados e tipagem estática;
- Sistema de tipagem sólido;
- Fornece isolamento em contextos de concorrência para permitir a existência de workers independentes que não compartilhem memória;
- Executa a compilação antecipada para o código nativo a fim de atingir os mais altos níveis de desempenho, equivalentes aos do Assembly;
- Dá suporte a um vasto repositório de pacotes que fornece funcionalidades adicionais;
- Há o suporte a diversas ferramentas de desenvolvedor populares que incluem o Visual Studio Code e o IntelliJ IDEA;

Desenvolvimento de aplicativos móveis

Widgets para tudo!

- No Flutter, tudo é widget! (“Quase”)
 - Widgets: bloco de códigos ou bloco de UI (“quase tudo UI”)
 - Ex.:
 - » `Text(“Hello!”)`
 - » `RaiseButton(`
 `onPress : function() {`

Desenvolvimento de aplicativos móveis

Widgets para tudo!

- Hierarquia de widgets:
 - Um widget Center,
 - com um widget Container abaixo dele,
 - esse widget Container com um widget Row embaixo,
 - e dois filhos Text sob Row além de um RaisedButton.
- Toda a hierarquia é um widget
- No Flutter, itens que normalmente não consideramos widgets também são widgets:
 - Preenchimento (*padding*) ao redor de uma imagem;
 - Estado do campo de texto de um formulário;
 - Texto exibido na tela;
 - Tema que um aplicativo em uso.

```
Center(  
  child : Container(  
    child : Row(  
      Text("Child 1"),  
      Text("Child 2"),  
      RaisedButton(  
        onPressed : function() {  
          // Faz algo  
        },  
        child : Text("Click me")  
      )  
    )  
  )  
)
```

ÁRVORE DE WIDGETS

Desenvolvimento de aplicativos móveis

Widgets para tudo!

- Quase todos os widgets são contêineres.
 - Logo, eles podem ter filhos;
- Todos os widgets são classes Dart e normalmente têm um único requisito:
 - Devem fornecer um método **build()**.
 - Esse método retorna outros widgets;
 - Algumas exceções retornam tipo primitivo;

Flutter

Vantagens:

- Hot reload;
- Multiplataforma;
- Dart;
- Widgets: veem com rico conjunto de widgtes;
- Ferramentas: é fácil criar uma configuração básica de ambiente de desenvolvimento para o Flutter;
- Widgets específicos da plataforma: uma única codebase que dê suporte tanto ao iOS quanto ao Android;

Flutter

Desvantagens:

- Mistura de códigos: mistura front-end e back-end;
- Google (Pode ser dependendo do ponto de vista);
- A árvore de widgets: pode ser difícil depurar o código principalmente em hierarquia aninhada (devido a profundidade na árvore);
- Programação reativa e gerenciamento de estado;
- Tamanho do aplicativo;