

# ISW-011

## Desenv. para Dispositivos Móveis 1 (DDM1)

DDM-1 – ISW-011  
Prof. Davi dos Reis

Aula 03 – Conceitos iniciais

### Conceitos Iniciais

## Conceitos Iniciais – características gerais

- Tamanho limitado;
- Memória parcialmente limitada;
- Processamento parcialmente limitado;
- Consumo de energia limitado;
- Conectividade limitada;
- Operam em ambiente hostil;
- Tempo de resposta deve ser rápido.

## Conceitos Iniciais – Desenvolvimento Nativo x Híbrido

- **Desenvolvimento Nativo:**
  - Um aplicativo é nativo quando programado em linguagem exclusiva do sistema (com uso de sua SDK). Por exemplo, um aplicativo nativo para Android usa-se o Java (ou Kotlin) como linguagem. Já para o iOS a linguagem Swift.
- **Desenvolvimento Híbrido:**
  - Um aplicativo híbrido, por sua vez, não é desenvolvido dentro da linguagem específica de cada sistema operacional: é feito com base em HTML5, CSS3 e JavaScript (alocado num container, que integra as funcionalidades com o S.O.).
  - Um app híbrido apresenta várias linguagens que englobam dois formatos: o próprio nativo e o web. Sendo assim, esse modelo pode utilizar recursos tanto da internet quanto do dispositivo.

## Conceitos Iniciais – Vantagens de cada formato

- **Desenvolvimento Nativo – Vantagens:**
  - Possibilita explorar melhor toda a parte de UI/UX, de acordo com o S.O.
  - Possuem integração direta com as bibliotecas e ecossistema da respectiva plataforma.
  - Geralmente, as atualizações de plataforma e novos recursos são suportados primeiramente/mais rapidamente no desenvolvimento nativo.
  - Tendem a ser mais rápidos e confiáveis que os demais tipos, por serem otimizados para cada plataforma\*.
  - Podem utilizar diretamente as funcionalidades do dispositivo e com bom desempenho (como câmera, acelerômetro, geolocalização etc.)\*.

Prof. Davi

ISW-011 (DDM-1)

6

## Conceitos Iniciais – Vantagens de cada formato

- **Desenvolvimento Híbrido/*Cross-Plataform* – Vantagens:**
  - Custo e tempo de desenvolvimento menores, pois uma única base de código gera aplicativos para plataformas diferentes (2, 3 ou até 4).
  - Menor curva de aprendizado (e menor tempo para desenvolvimento): basta aprender ou usar o conhecimento em uma única linguagem para criar apps em plataformas diferentes.
  - Maior capacidade de entrega com equipe menor (iOS, Android e web).
  - Manutenções podem ser implementadas de uma única vez para ambas as plataformas, reduzindo carga de trabalho.
  - Possibilidade de prototipagem e iterações rápidas, sendo bastante interessante para startups, testes de hipóteses, projetos com prazos exíguos etc.

Prof. Davi

ISW-011 (DDM-1)

7

## Conceitos Iniciais

~ Android e iOS ~



Prof. Davi

ISW-011 (DDM-1)

8

## Android

- Plataforma padrão para desenvolvimento de aplicativos para smartphones, smartwatches, smartTVs, smartCards e tablets;
- A plataforma tem diversos atrativos:
  - Programação de aplicações usando Java/Kotlin;
  - Programação de aplicações usando C++ (nativo);
  - Gratuita;
  - Código-fonte aberto;
  - Flexível.



Prof. Davi

ISW-011 (DDM-1)

9

# Android

- Não é apenas Google!
- Com outras empresas, formam a Open Handset Alliance:  
<http://www.openhandsetalliance.com>
- Google, HTC, Intel, Samsung, Motorola, etc.
- O objetivo desse consórcio é buscar um padrão aberto que atenda as necessidades atuais.



Prof. Davi

ISW-011 (DDM-1)

10

## Open Handset Alliance



Operator	Handset Makers	Software Companies	Commercialization Companies	Semiconductor Companies

Prof. Davi

ISW-011 (DDM-1)

11



## Android – versões

- 1.0 Astro (2008);
- 1.5 Cupcake (2009);
- 1.6 Donut (2009);
- 2.0 2.1 Eclair (2010);
- 2.2 Froyo (2010);
- 2.3 Gingerbread (2010);
- 3.0 3.2 Honeycomb (Tablets);
- 4.0 Ice Cream Sandwich (Tablet e Smartphone);
- 4.1 4.2 4.3 Jelly Bean;
- 4.4 Kitkat;
- 5.0 Lollipop;
- 6.0 Marshmallow;
- 7 Nougat;
- 8 Oreo;
- 9 Pie;
- Android 10;
- Android 11;
- Android 12;  
<https://developer.android.com/about/versions/12>
- Android 13;  
<https://developer.android.com/about/versions/13>
- Android 14;  
<https://developer.android.com/about/versions/14>
- Android 15 (Preview);  
<https://developer.android.com/about/versions/15>

Prof. Davi

ISW-011 (DDM-1)

12

## Android – Ambiente de Desenvolvimento

- Sistema Operacional:
  - Windows, Linux, MacOS;
- Ferramentas de Desenvolvimento:
  - Android Studio e SDK;
  - Linguagens Kotlin, Java e C++.



Prof. Davi

ISW-011 (DDM-1)

13

## iOS

- Plataforma padrão para desenvolvimento de aplicativos para smartphones, smartwatches, smartTVs, smartCards e tablets;
- Solução proprietária da Apple;
- A plataforma tem diversos atrativos:
  - Leveza
  - Fluidez
  - Rapidez
  - Padronização.



## iOS – versões

- Phone OS 1 (2007 – lançamento do iPhone original);
  - iPhone OS 2 (2008);
  - iPhone OS 3 (2009);
  - iOS 4 (2010);
  - iOS 5 (2011) – inclusão do armazenamento iCloud;
  - iOS 6 (2012);
  - iOS 7 (2013) – primeira versão com a assistente virtual Siri;
  - iOS 8 (2014) – apresentação da compatibilidade com o Apple Watch;
  - iOS 9 (2015);
  - iOS 10 (2016);
  - iOS 11 (2017);
  - iOS 12 (2018);
  - iOS 13 / iPadOS 13 (2019);
  - iOS 14 (2020);
  - iOS 15 (2021);
  - iOS 16 (2022);
  - iOS 17 (2023);
- <https://www.apple.com/ios/ios-17/>



## iOS – Ambiente de Desenvolvimento

- Sistema Operacional:
  - MacOS;
- Ferramentas de Desenvolvimento:
  - Xcode;
  - Objective-C (inicialmente);
  - Linguagem Swift.



## Iniciativas low/no-code



## Iniciativas low/no-code

- MIT – App Inventor 2  
<https://appinventor.mit.edu/>
- Kodular  
<https://www.kodular.io/>
- Thunkable  
<https://thunkable.com/>
- Glide  
<https://www.glideapps.com/>

## Frameworks de desenvolvimento

## Frameworks de Desenvolvimento



▪ Flutter  
<https://flutter.dev/>



▪ React Native  
<https://reactnative.dev/>



▪ Ionic  
<https://ionicframework.com/>



▪ Xamarin  
<https://dotnet.microsoft.com/en-us/apps/xamarin>

## Frameworks – comparativo

Flutter	React Native	Ionic
Dart + Flutter	Javascript / React.js	Javascript (qualquer framework)
App Nativa Compilada	App Nativa Parcialmente Compilada	App dentro de uma WebView
NÃO usa componentes nativos do iOS/Android	Utiliza componentes nativos do iOS/Android	NÃO usa componentes nativos do iOS/Android
Cross-Platform (Mobile, Web e Desktop)	Foco em Mobile	Cross-Platform (Mobile, Web e Desktop)
Desenvolvido pelo Google	Desenvolvido pelo Facebook	Desenvolvido pelo Ionic

## Prática!! (preparando o ambiente)

Prof. Davi

ISW-011 (DDM-1)

28

## Instalação – Softwares necessários:

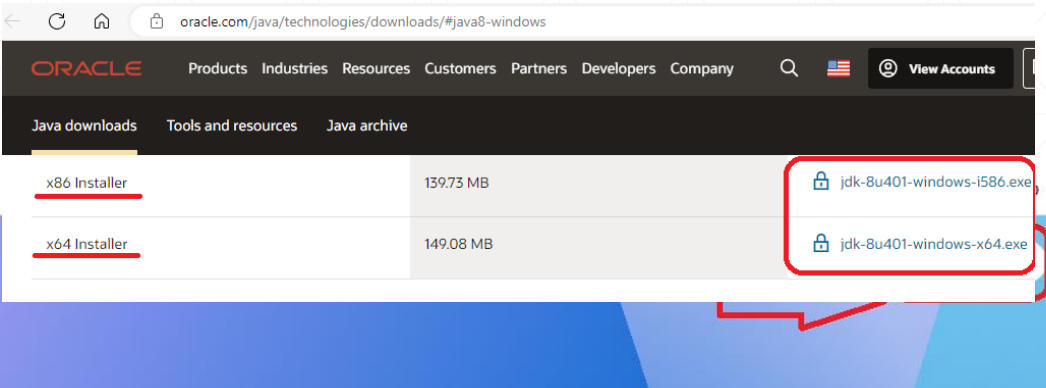
- **Android Studio** (apenas para uso da SDK Android + Emulador)  
<https://developer.android.com/studio?hl=pt-br>
- **Visual Studio Code (VSCode)**  
<https://code.visualstudio.com/download>
- **Flutter**  
<https://flutter.dev/>
- **JDK do Java** (no caso, Java SE 8 (JDK Java 8))  
<https://www.oracle.com/java/technologies/downloads/#java8-windows>

Prof. Davi

ISW-011 (DDM-1)

29

# Framework Flutter – Instalação



Prof. Davi

ISW-011 (DDM-1)

30

# Framework Flutter – Instalação

- Após isso, ainda no “C:\src\flutter\”, executar o comando **flutter doctor -v**, para verificar se você já tem tudo instalado na sua máquina para desenvolvimento Flutter ou se tem alguma pendência.
- Possivelmente haverá pendência sobre licenças; neste caso, deve-se rodar o comando “**flutter doctor --android-licenses**” para aceitar as licenças Android.
- Rodar novamente o “flutter doctor -v” até não haver pendências.
- Se já estiver com o VS Code instalado, **instalar** as **extensões** para uso do **Dart** e do **Flutter** (que já possuem esse nome).

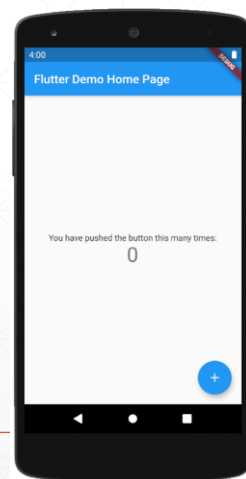
Prof. Davi

ISW-011 (DDM-1)

31

## Framework Flutter – Teste de ambiente

- No Visual Studio Code (**VSCode**):
  - Para testar o ambiente, vamos criar um novo projeto
  - Para isso, utilize as teclas de atalho **CTRL + SHIFT + P**
  - Escolha **Flutter: New Application Project** e nomear o projeto
  - Um projeto padrão será criado. Ao executá-lo, será possível ver uma tela semelhante à da imagem ao lado.
  - **Parabéns!** Primeiro projeto Flutter no VS Code!



Prof. Davi

## Framework Flutter e Ling. Dart – Instalação

- **Linguagem Dart**
  - Para saber mais:
    - <https://www.dartlang.org/guides/language/language-tour>
    - <https://dart.dev/get-dart>
  - Por ora, para execução no laboratório, poderemos utilizar apenas a versão online:
    - <https://dartpad.dartlang.org/> ou <https://dartpad.dev/> (editor online da linguagem)
- **Vysor** (opção para espelhar a tela do celular no computador)
  - <https://www.vysor.io/>
  - Sobre o Vysor: <https://www.youtube.com/watch?v=qAHTini67GA&t=24s>

Prof. Davi

ISW-011 (DDM-1)

33

# Hello, World! (Dart)

## Dart – Projeto inicial padrão

```
void main() {  
  for (int i = 0; i < 5; i++) {  
    print ('hello ${i + 1}');  
  }  
}
```

VS Code  
<https://dartpad.dev/>  
<https://dartpad.dartlang.org/>

# Primeiro Projeto – “Hello, world”

```
void main() {  
  
    final String msg = "Hello, world!";  
  
    // Comentário em linha  
    print("Exibirá a primeira mensagem!!");  
    print("Mensagem: $msg");  
}
```

<https://dartpad.dev/>  
<https://dartpad.dartlang.org/>

Obrigado!