

Aula 1 – Introdução ao desenvolvimento Mobile

Professor Bruno Maion

Introdução

- **Aplicativos Nativos**

- Desenvolvidos para uma plataforma específica (Android, iOS).
- Linguagens: Kotlin/Java (Android), Swift/Objective-C (iOS).
- Acesso total a recursos do dispositivo (GPS, câmera, sensores).
- Maior performance e responsividade.
- Exigem manutenção separada por plataforma.

Introdução

- **Aplicativos Web (Web Apps)**
 - Executados no navegador (não precisam de instalação).
 - Desenvolvidos com HTML, CSS, JavaScript.
 - Responsivos (adaptam-se a diferentes tamanhos de tela).
 - Não acessam recursos avançados do dispositivo facilmente.
 - Boa portabilidade, mas com limitações de performance.

Introdução

- **Aplicativos Híbridos**

- Combinação de código web com contêiner nativo.
- Ex: Ionic, Cordova, Capacitor.
- Compartilham código entre plataformas.
- Maior agilidade de desenvolvimento, mas com possível perda de desempenho.
- Acesso limitado aos recursos nativos, dependendo da ferramenta usada.

Introdução

Característica	Nativo	Híbrido	Web App
Desempenho	Excelente	Médio	Baixo/médio
Acesso a hardware	Completo	Parcial	Limitado
Manutenção	Separada por SO	Compartilhada	Única (web)
Instalação	Sim	Sim	Não
Custo	Alto	Médio	Baixo

Conceitos – *Server-Side* e *Client-Side*

Server-side

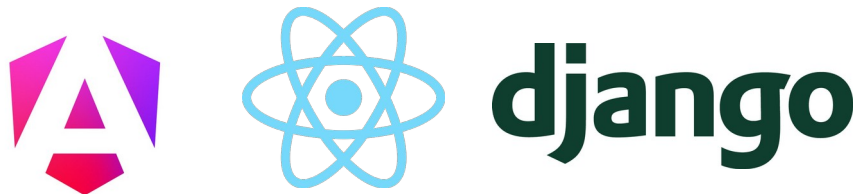
- Executado no **servidor**;
- Gera o conteúdo da página **antes** de enviá-la ao navegador;
- Usa linguagens como PHP, Python, **Node.js**;
- **Acesso direto** ao banco de dados e lógica de negócio;
- **Mais seguro**, pois o código não é exposto ao usuário.

Client-side

- Executado no navegador do **usuário**;
- Manipula a interface **após** o carregamento da página;
- Usa **JavaScript, HTML, CSS**;
- Ideal para validações, animações e interatividade;
- **Menos seguro**, pois o código é visível;

Conceitos - Framework

- Um **framework** é uma estrutura base de código e ferramentas que facilita o desenvolvimento de software.
- **Bibliotecas** oferecem **funções específicas** que você chama; **frameworks** definem a estrutura e chamam o seu código dentro dela.
- Fornece **componentes** prontos, padrões de arquitetura, e regras que guiam o programador.
- Em vez de escrever tudo do zero, o desenvolvedor reutiliza e estende funcionalidades do framework
 - **Web:** Angular, React, Vue.js.
 - **Mobile:** React Native, Flutter, Ionic.
 - **Backend:** Django, Laravel, Express.



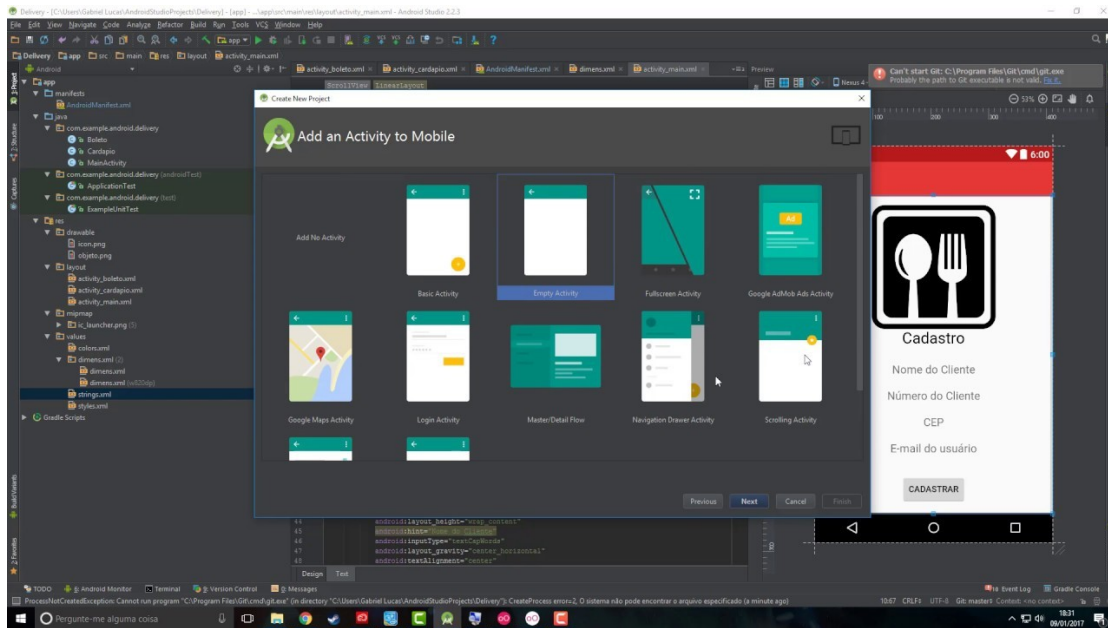
Conceitos - Build

- O processo de **build** (ou construção) é a etapa que transforma o código-fonte em um produto executável.
- **Compilação** do código (ex: transformar TypeScript em JavaScript ou Dart em binário).
- **Empacotamento** de recursos (imagens, estilos, bibliotecas).
- Otimização do código.
- Geração de arquivos **finais** como .apk, .ipa, .exe, ou .zip.
- Pode ser feito localmente, com ferramentas como Webpack, Gradle, Xcode.
- Na nuvem, via serviços de **CI/CD** (ex: GitHub Actions, EAS Build, App Center).

 *"Build é a **fábrica** que monta o produto final a partir do código"*

Ambientes Nativos – Android (Google)

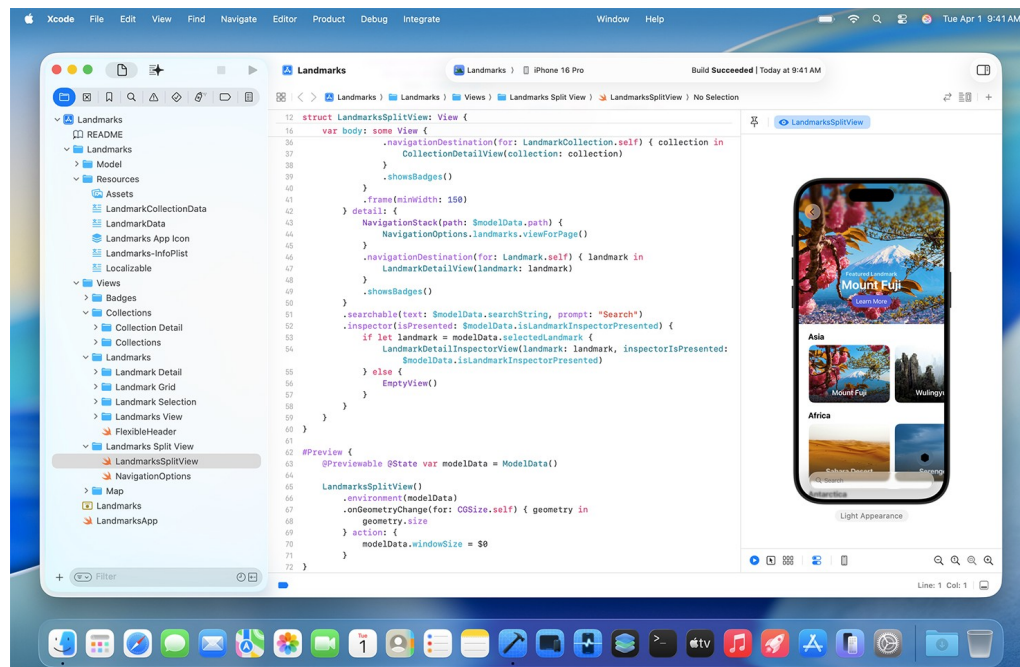
- Sistema Operacional: **Android** (baseado em **Linux**).
- Linguagens oficiais:
 - **Kotlin** (recomendada pela Google).
 - **Java** (ainda amplamente utilizada).
- Ambiente de Desenvolvimento (**IDE**): Android Studio (baseado no IntelliJ IDEA).
- **SDKs**: Android SDK (Software Development Kit).
- Ferramentas como ADB (Android Debug Bridge), emuladores e **build** system (**Gradle**).
- **Distribuição**: Publicação via **Google Play Store**.



Android Studio

Ambientes Nativos – iOS (Apple)

- Sistema Operacional: **iOS** (baseado em **Unix**).
- Linguagens oficiais:
 - **Swift** (moderna e segura).
 - **Objective-C** (herança de versões anteriores).
- Ambiente de Desenvolvimento (IDE): **Xcode** (oficial da Apple).
- **SDKs**: iOS SDK, com frameworks como UIKit, SwiftUI e ARKit.
- Distribuição: Publicação via **Apple App Store** (com validação rigorosa).



Xcode

Ambiente Web

Node.js: Ambiente de execução JavaScript no servidor

- Gerencia dependências (via NPM/Yarn).
- Executa scripts de build, testes e automações.
- Usado como backend ou suporte ao ambiente de desenvolvimento.

React Native: Framework

- Criado pela Meta (Facebook) para desenvolvimento mobile usando React.
- Escreve-se em JavaScript ou TypeScript.
- Gera aplicativos nativos, com componentes que se convertem para Android/iOS.

React vs React Native

React (ou ReactJS // de Javascript)

- Biblioteca **JavaScript** para criar interfaces web.
- Executa no navegador (**client-side**).
- Usa **HTML**, CSS e JavaScript/JSX.
- Componentes são convertidos em **DOM** HTML.
 - é a representação em árvore de uma página HTML.
- Aplicações são **SPA** (Single Page Applications).
 - Tipo de aplicação web que carrega uma **única** página HTML e atualiza seu conteúdo **dinamicamente**, sem **recarregar** toda a página do navegador.
- Ex: Sites, painéis administrativos.

React Native

- **Framework** baseado em **React** para **apps** móveis nativos.
- **Executa** no dispositivo (**Android/iOS**).
- Usa **JavaScript/JSX**, mas não usa HTML/CSS.
- Componentes são convertidos em **widgets** nativos (ex: **<View>**, **<Text>**, **<Button>**).
 - **Componentes** (unidade reutilizável) de interface gráfica fornecidos pelo sistema operacional, como botões, campos de texto e listas, com aparência e comportamento nativos da plataforma (Android/iOS).
- Permite escrever um **único** código para Android e iOS.
- Exemplo de uso: aplicativos como Instagram, Uber Eats, Facebook.



Obrigado!