



Programação para Dispositivos Móveis

OVERVIEW DO DESENVOLVIMENTO MOBILE

Professor Jean Carlo Wagner

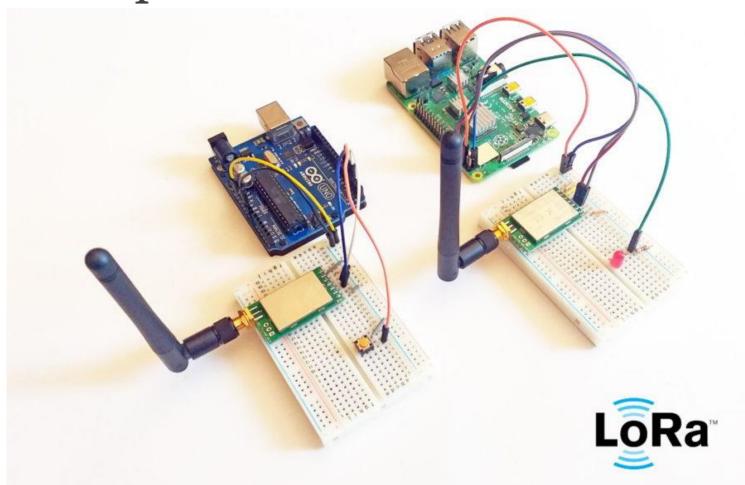
Agenda

- > Dispositivos móveis: conceito
- > Principais aplicações nativas
- ➤ Indústria 4.0
- > Cross-platform
- > Apps híbridos
- > Pensamentos para uma solução ...
- > Características
- > Alguns frameworks para desenvolvimento

Dispositivos móveis: conceito

- Um dispositivo móvel, designado popularmente em inglês por handheld, é um "computador" de bolso, habitualmente equipado com uma pequena tela (output) e um teclado em miniatura (input). Em muitos aparelhos, o output e o input combinam-se num ecrã tátil (touchscreen).
- Sua principal característica é a comunicação entre equipamentos, com mobilidade. Além disto, seu poder computacional aumenta deste seu surgimento, possibilitando este mercado de aplicativos, presente em nosso cotidiano.

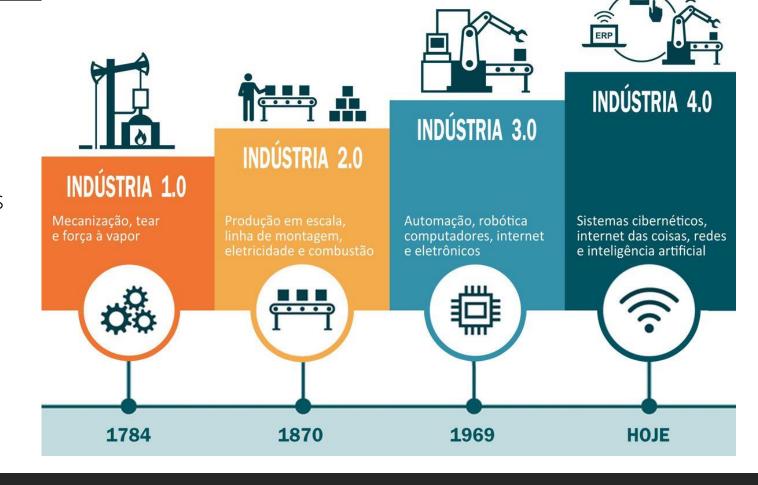
Dispositivos móveis: conceito





Indústria 4.0

Indústria 4.0 ou Quarta Revolução Industrial é uma expressão que engloba algumas tecnologias para automação e troca de dados e utiliza conceitos de sistemas ciberfísicos, Internet das coisas e computação em nuvem. O foco da quarta Revolução Industrial é a melhoria da eficiência e produtividade dos processos.



Principais aplicações nativas

> Android

- Originalmente em Java
- Hoje permite o uso de Kotlin
- Desenvolvimento acontece 100% na IDE Android Studio











- Originalmente em Objective-C
- Hoje permite o uso de Swift
- Desenvolvimento acontece 100% na IDE XCode









Cross-platform

- E se pudéssemos usar o mesmo código para ambos os apps?
- Hoje temos frameworks que compilam em app nativo, como as do quadro ao lado.
- Embora a performance não seja a mesma de apps realmente nativos, a diferença é imperceptível na maioria dos casos.

PROGRAMMING LANGUAGE	CROSS-PLATFORM APP DEVELOPMENT FRAMEWORK
3	Xamarin
Java	Codename One
©	Qt
J5	Appcelerator, NativeScript, Cordova, Ionic, React Native, JavaScript framework
- python	Kivy, BeeWare
Ruby	RubyMotion
Dart	Flutter
Visual Basic	B4A

Apps híbridos

➤ No entanto estamos esquecendo de uma plataforma muito importante: E se nós também quisermos que nosso *app* funcione na *web*? Na grande maioria dos casos é desejável rodar em *desktops/laptops/notebooks* e ter uma versão *web* responsiva que funcione no *mobile*, mesmo que com menos funcionalidades.

> Problemas:

Em *apps* nativos, teríamos 3 equipes: (1) iOS; (2) Android; (3) *web*; Em *apps cross-platform*, teríamos 2: (1) *mobile*; (2) *web*;

Em poucos casos essa separação é justificável!

Pensamentos para uma solução ...

- Escrever uma aplicação como se fosse o front-end de um site, em HTML, CSS e JavaScript.
- Apps para Android e iOS possuem um componente chamado WebView, o qual permite inserir conteúdo HTML, CSS e JavaScript.
- Usar uma ferramenta que utilize o front-end inteiro e simplesmente envolva-o dentro de um app cuja única funcionalidade é apresentar um WebView.
- Além disso, essa ferramenta proporciona ao app acesso às funcionalidades nativas de uma forma unificada.

Native App with a WebView

Web Content

 $\Lambda \Psi$

Native App

WebView

Web Rendering Engine

JavaScript Bridge

Native Bridge

 \uparrow

Native APIs / Platform

Web App

Web Content



Browser

Shell / UI

Web Rendering Engine



Native APIs / Platform



... e além disto ...



Características

- Finalmente temos um único codebase que contempla mobile e web.
- Podemos fazer nosso app se comportar diferente dependendo de onde está rodando, caso necessário e assim ganhamos novas plataformas: os PWAs¹ e aplicações desktop nativas (com Electron, por exemplo).
- A performance tende a ser um pouco mais baixa do que os nativos e crossplatform, mas costuma ser mais que satisfatória.



> React Native

• É um dos principais frameworks cross-platform e proporciona recursos baseados em Java. Como é um framework de código aberto, possui uma vasta comunidade de desenvolvedores. Alguns exemplos usando esse framework são Airbnb, Skype, Instagram, Uber Eats, Pinterest, entre outros.



> Xamarin

- É um framework de desenvolvimento de aplicativos de código aberto que foi criado pela primeira vez em 2011. Era inicialmente independente, mas foi adquirido pela Microsoft em 2016. A cross-platform Xamarin é baseada na execução de código aberto da plataforma .NET, conhecida como Mono. Os componentes do processo de execução incluem bibliotecas .NET, compilador C# e tempo de execução.
- Xamarin tem uma enorme comunidade de desenvolvedores e abrange plugins e compiladores do Visual Studio para iOS e Android. Alguns aplicativos populares criados com essa estrutura incluem Alaska Airlines, BBC Good Food, HCL, Fox Sports, entre outros.



> Flutter

- Esse *framework* foi lançado em maio de 2017 pelo Google como um dos principais *frameworks* de desenvolvimento multiplataforma.
- Utiliza linguagem de programação Dart em vez de JavaScript. Os desenvolvedores de aplicativos podem alterar o código e visualizar as alterações em tempo real devido ao recurso hot reload. Além disso, a linguagem de programação Dart permite que os desenvolvedores de aplicativos escrevam códigos de maneira estruturada para aplicativos complexos.
- Tem seu próprio mecanismo gráfico, evitando com que os desenvolvedores criem uma interface de desenvolvimento de aplicativos Android e iOS separada. Alguns aplicativos famosos usando Flutter incluem Alibaba, My Leaf, Google Ads, entre outros.

> Ionic

• É um *framework* multiplataforma referenciado em CSS, JavaScript e HTML. Além disso, recebe *plugins* suficientes para usar os recursos móveis nativos sem problemas. Os aplicativos móveis desenvolvidos por meio dessa estrutura são aplicativos HTML híbridos baseados nos princípios do Apache Cordova. Os principais aplicativos móveis desenvolvidos usando esse *framework* incluem Instant Pot, Pacifica, Shipt, Sworkit, Honeyfi, entre outros.



➤ Native Script

- O Native Script torna desnecessário que os desenvolvedores de aplicativos usem WebView, pois fornece uma interface usuário nativa acessível e possui recursos dinâmicos. Além disso, oferece todas as APIs permitindo que os desenvolvedores reutilizem os plugins atuais do NPM para os projetos.
- Os desenvolvedores de aplicativos não precisam conhecer as linguagens nativas de desenvolvimento de aplicativos por meio do React Native; podem acessar APIs do Android e iOS.
- Native Script também utiliza Angular e TypeScript para tarefas relacionadas à programação.



Dúvidas!?

