



PADO
Labs



Desenvolvimento Mobile

Henrique Carlos



PADO
Labs

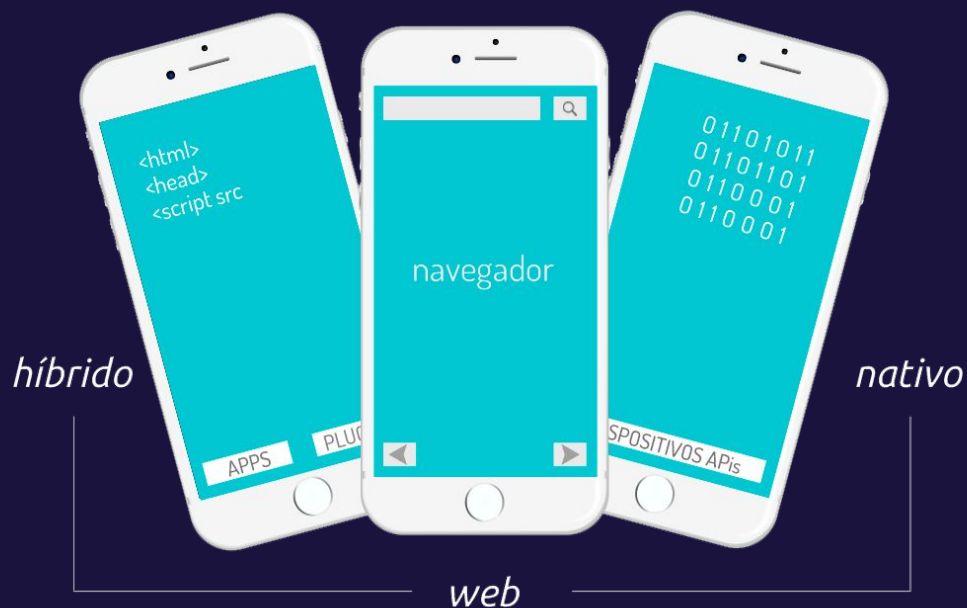
Aula 1

Introdução Ionic e Capacitor

Tipos de aplicativos



APPS NATIVO, WEB E HÍBRIDO



Apps Nativos



Android (JAVA / Kotlin) – IOS (Swift / Objective-c)

Os apps nativos são desenvolvidos com as linguagens dos próprios sistemas operacionais, podendo utilizar funcionalidades como câmera, GPS, entre outras. Além disso, boa parte deles funciona offline.

Para que sejam disponibilizados aos usuários, precisam atender aos requisitos das lojas, como normas de segurança e privacidade.

Vantagens

- Mais recursos
- Mais fluidez do App
- Menos bugs
- Mais Segurança

Desvantagens

- Necessário dois app
- Desenvolvimento mais lento
- Maior custo de desenvolvimento

Apps Híbridos



Capacitor, React Native, Flutter

O app híbrido é desenvolvido em linguagens web, mas se apresenta e se comporta como um aplicativo nativo — que utiliza linguagem específica —, funcionando como uma união entre os dois modelos e sendo inclusive instalado em lojas de apps (como a App Store e a Play Store).

Vantagens

- Menor Custo
- Desenvolvimento mais rápido
- Desenvolvimento único

Desvantagens

- Recursos Limitados
- Baixa Performance

Apps Webs (PWA)



HTML, CSS

Essa categoria é nada mais do que um site responsivo, que se comporta como um aplicativo. Ele pode ser acessado de qualquer tipo de navegador e funciona nos mais diversos tamanhos de tela.

O acesso dele é através de uma URL e o usuário tem a opção de adicionar o ícone do site na tela inicial do aparelho, como se fosse um aplicativo baixado direto da loja.

Vantagens

- Não precisa colocar nas lojas
- Não necessita um novo código

Desvantagens

- Sem recursos do celular
- Necessário internet
- Pouco uso

Capacitor & Ionic



Capacitor é uma plataforma que facilita a criação de aplicativos de alto desempenho que são executados nativamente em iOS, Android e muito mais usando ferramentas modernas da Web. Representando a próxima evolução dos aplicativos híbridos, o Capacitor cria aplicativos nativos da Web, fornecendo uma abordagem moderna de contêiner nativo para equipes que desejam criar primeiro na Web sem sacrificar o acesso total a SDKs nativos quando necessário.

O Ionic é um kit de ferramentas de interface do usuário de código aberto para criar aplicativos móveis de alta qualidade e desempenho usando tecnologias da Web — HTML, CSS e JavaScript — com integrações para estruturas populares como Angular, React e Vue.

Ferramentas



- Android Studio
- Android SDK
- Android Platform Tools
- VSCode

- Instalar NODE.JS
<https://nodejs.org/en/>

* NPM (Node Package Manager)

Ir até o CMD e checar se a instalação deu certo.

- node -v

Instalar IONIC CLI (Command Line Interface)

- Npm install -g @ionic/cli

Iniciar projeto



- Ionic start

Use the App creation Wizard ?

N

Framework:

Angular

Project Name:

Hello Word

Starter Template:

Blank

Creat Free Account ?

N

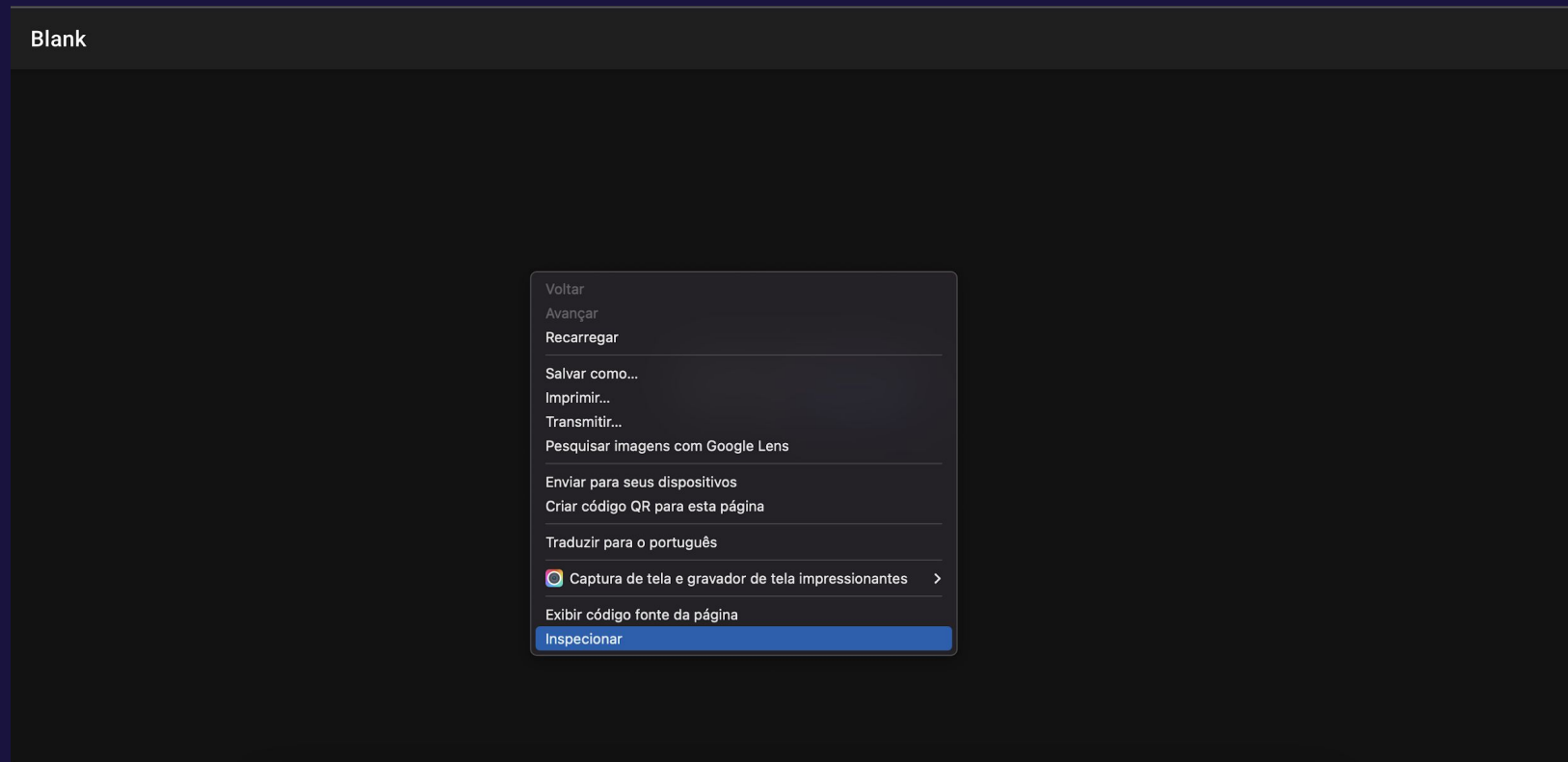
- Cd hello-word
- Ionic capacitor add
- Android
- Ionic capacitor add
- IOS

Como rodar o Projeto



- Ionic serve

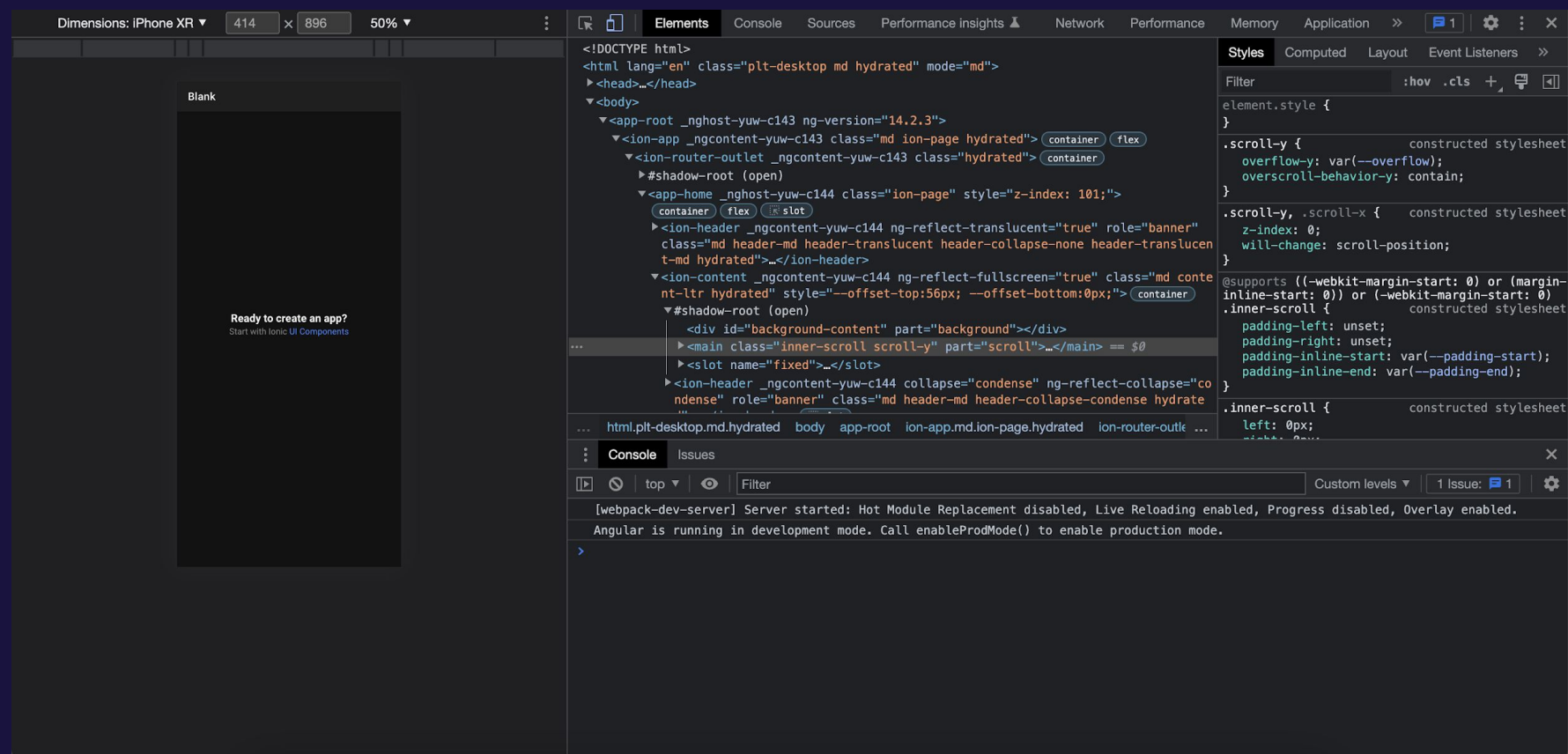
- Abrir URL no navegador <http://localhost:8100/home>
- Com o botão direito do mouse, abrir o Inspeccionar elemento.



Como rodar o Projeto



- Selecione o segundo elemento do menu superior para simular uma tela de celular.



Estrutura



- E2E (usado para testes, não iremos utilizar)
- Node_modules (Contém bibliotecas)
- Src (Contém todo o código do app)
 - App
 - Assest
 - Enviroments
 - Theme
- Tipos de Arquivos
 - Html
 - Scss
 - Specs.ts (testes)
 - ts

App



Quando iniciamos o projeto automaticamente o Ionic cria uma página inicial, essa página é utilizada como default. Essa tela vem em branco, e nela podemos fazer o que quisermos utilizando os componentes do Ionic.

<https://ionicframework.com/docs/components>

Os aplicativos Ionic são feitos de blocos de construção de alto nível chamados Componentes, que permitem que você construa rapidamente a interface do usuário do seu aplicativo.

O Ionic vem com vários componentes, incluindo cartões, listas e guias. Quando estiver familiarizado com o básico, consulte o Índice da API para obter uma lista completa de cada componente e subcomponente.

ion-card



- Os cartões são uma parte padrão da interface do usuário que serve como ponto de entrada para informações mais detalhadas. Um cartão pode ser um único componente, mas geralmente é composto por algum cabeçalho, título, subtítulo e conteúdo. ion-card é dividido em vários subcomponentes para refletir isso.

```
<ion-card>
  
  <ion-card-header>
    <ion-card-subtitle>Destino</ion-card-subtitle>
    <ion-card-title>Londrina, PR</ion-card-title>
  </ion-card-header>
  <ion-card-content>
    Londrina é um município brasileiro localizado no estado do Paraná, na Região Sul do Brasil, distando 381 km da capital paranaense, Curitiba. É considerada pela CGU a cidade mais transparente do Paraná.
  </ion-card-content>
</ion-card>
```

Outros Exemplos



- `lon-alert`
- `lon-badge`
- `lon-breadcrumb`
- `lon-breadcrumbs`
- `lon-button`
- `lon-card`
- `lon-card—contente`
- `lon-card-header`
- `lon-checkbox`
- `lon-chip`
- `lon-datetime`
- `lon-datetime-button`
- `lon-picker`
- `lon-grid`
- `lon-col`
- `lon-row`
- `lon-icon`
- `lon-item`
- `lon-item-divider`
- `lon-input`
- `lon-textarea`

Atividade 1



- Monta Página igual a página abaixo, utilizando componentes, e CSS
- Link FIGMA
- shorturl.at/abcMN

Adicionar nova página



- Ionic generate page “nome”
- Link pra nova página (click)=“goToPage()”
- No arquivos .ts criar

```
goToPage(){  
  alert('próxima página');  
}
```

```
// this.router.navigate(['/cidade']);
```

```
Import { Router } from '@angular/router';
```

Adicionar no constructor (private router: Router)



PADO
Labs