

IONIC + ANGULAR

DESENVOLVIMENTO MOBILE HIBRIDO



Raphael Policena



`</>` | Desenvolvedor Fullstack SR



Hitss do Brasil



SI - IF Goiano - Urutai



Orizona - GO

phaelpolicena

raphael-policena

POI1c3na

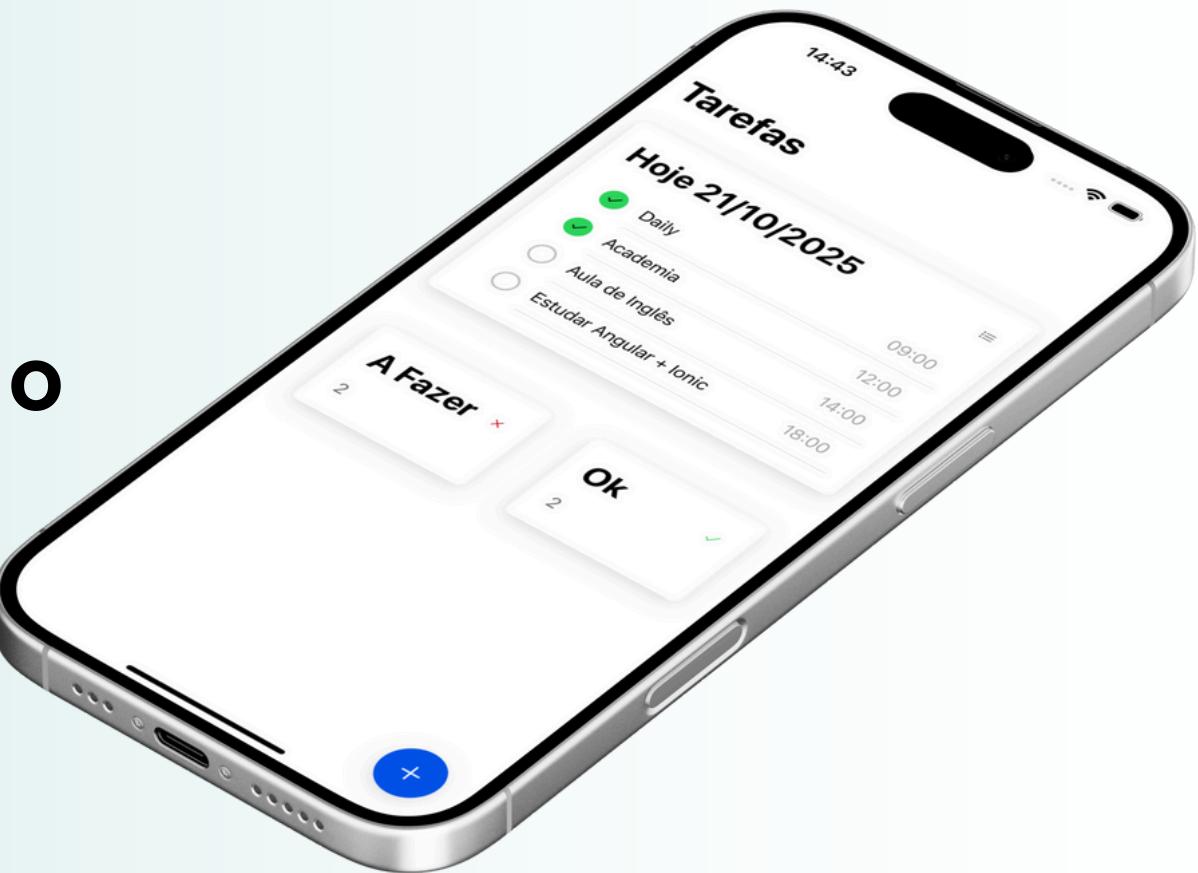
phaelpolicena@gmail.com

Programação

APP TO DO



1. Ionic
2. Angular
3. Instalação e criação de um novo projeto
4. Desenvolvimento do app To Do
5. Execução no Android e IOS
6. Atividades e Finalização





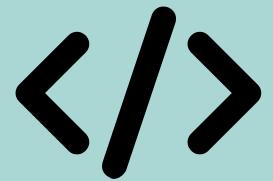
O que é o Ionic

- Um framework de desenvolvimento de aplicativos mobile híbridos
- Baseado em Angular
 - Primeira versão para AngularJS
 - Atualmente suporta Vue, React e frameworks JavaScript puros
- Criado em 2013 por Max Lynch, Ben Sperry e Adam Brendley

App híbrido

- Utiliza tecnologias Web
- Empacota e compila utilizando uma *Bridge* (Capacitor ou Cordova)
- O mesmo código executa em dispositivos móveis e no navegador
- Acessa recursos nativos via plugins (camera, storage, notifications)
- É exibido dentro de um WebView





Código Único

O mesmo código utilizado em várias plataformas



Menor custo

Menor custo de manutenção



Rápido

Por utilizarem a mesma base de código, sem a necessidade de desenvolvimento em outra linguagem



Vantagens e Desvantagens



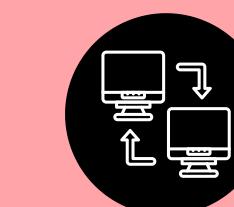
Desempenho inferior

Por conta do webview, o desempenho é ligeiramente inferior



Limitações

Dificuldades com APIs nativas complexas, como o Bluetooth BLE, libs que mexem com o DOM e etc.



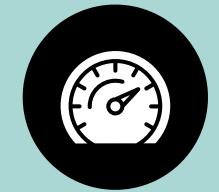
Ajustes

Pode precisar de ajustes visuais entre plataformas



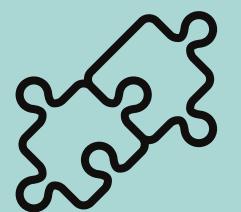
App nativo

- Desenvolvido para uma plataforma específica
 - Java/Kotlin para android
 - Swift ou Objective C para IOS



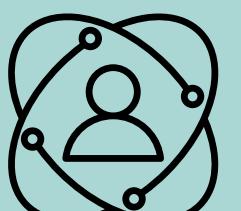
Desempenho

Melhor desempenho por executar e mostrar os elementos diretamente



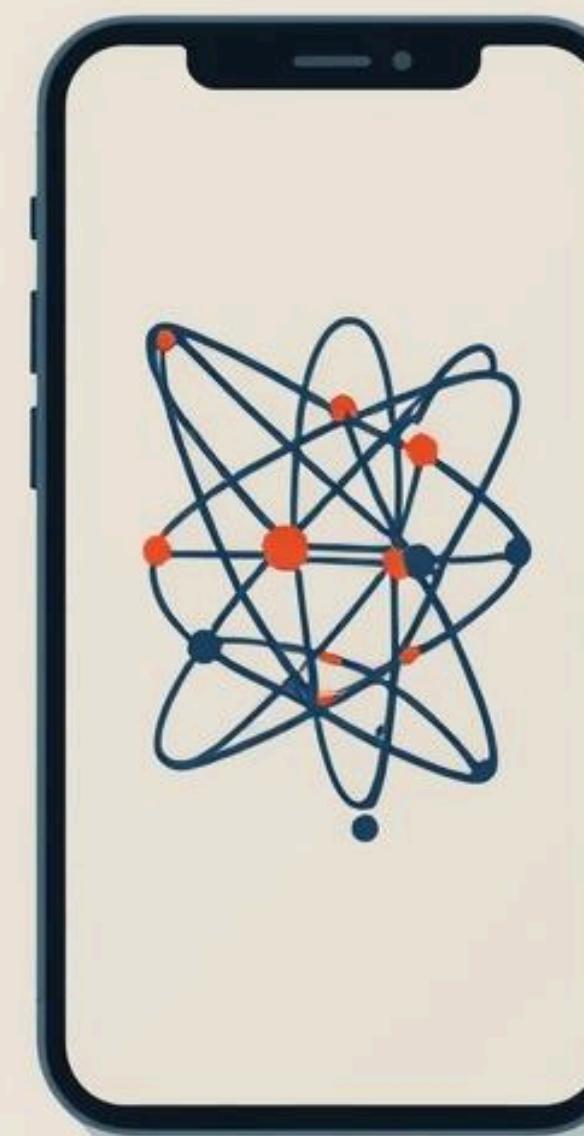
Integração

Acessa as APIs diretamente e com total acesso



Melhor experiência

Experiencia do usuário mais fluída e natural



Custo maior

Necessita de duas equipes, conhecimentos com específicos e dois projetos



Tempo

Maior tempo de desenvolvimento

Vantagens e Desvantagens

Estrutura

Um app Ionic é composto pelos diretórios principais:

- **app**
- **assets**
- **environments**
- **theme**

É recomendado a criação de diretórios dentro do diretório do app como models, pages e services

```
app-to-do-minicurso ~/Documents/Repos/Minicurso/app-to-do-minicurso
  > .angular
  > .idea
  > .vscode
  > node_modules library root
  > src
    > app
      > models
      > pages
      > services
      > app.component.html
      > app.component.scss
      > app.component.spec.ts
      > app.component.ts
      > app.module.ts
      > app-routing.module.ts
    > assets
    > environments
    > theme
      > global.scss
    > index.html
    > main.ts
    > polyfills.ts
    > test.ts
    > zone-flags.ts
```

Ionic LifeCycle

Ao entrar em
uma página

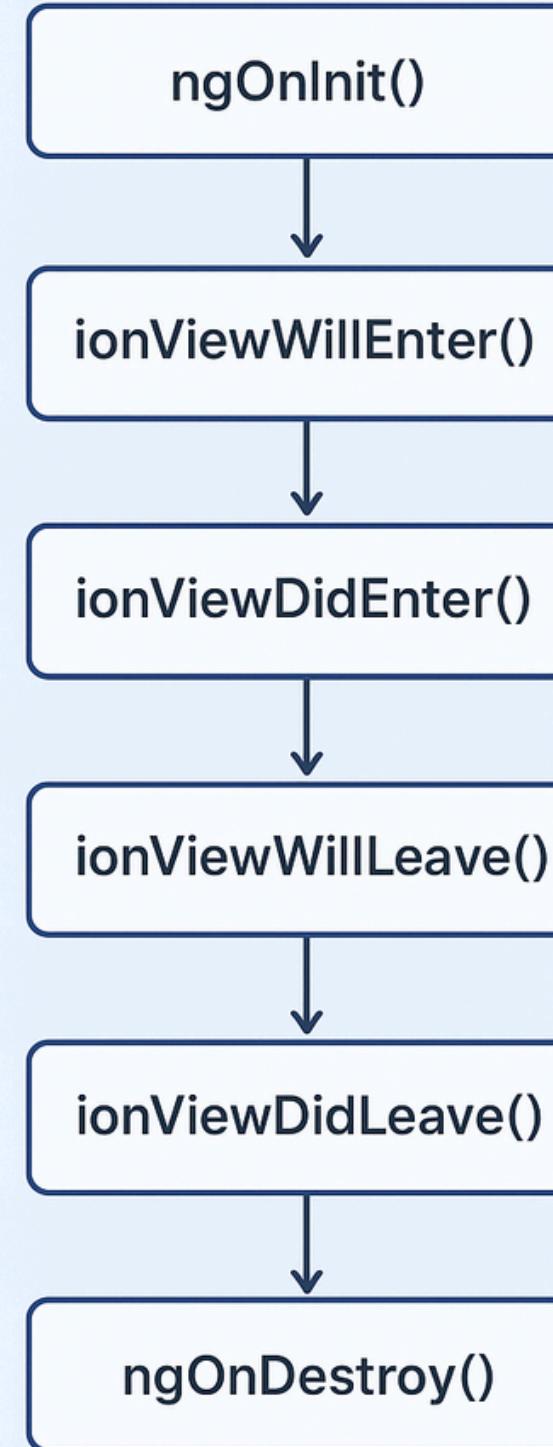
Prestes a
aparecer

Usando a página

Ao sair da página

Prestes a sair

Quando a página
é destruída



- Getting Started
- UI Components

- Accordion
- ion-accordion
- ion-accordion-group

- Action Sheet
- ion-action-sheet

- Alert
- ion-alert

- Badge
- ion-badge

- Breadcrumbs
- ion-breadcrumb
- ion-breadcrumbs

- Button
- ion-button
- ion-ripple-effect

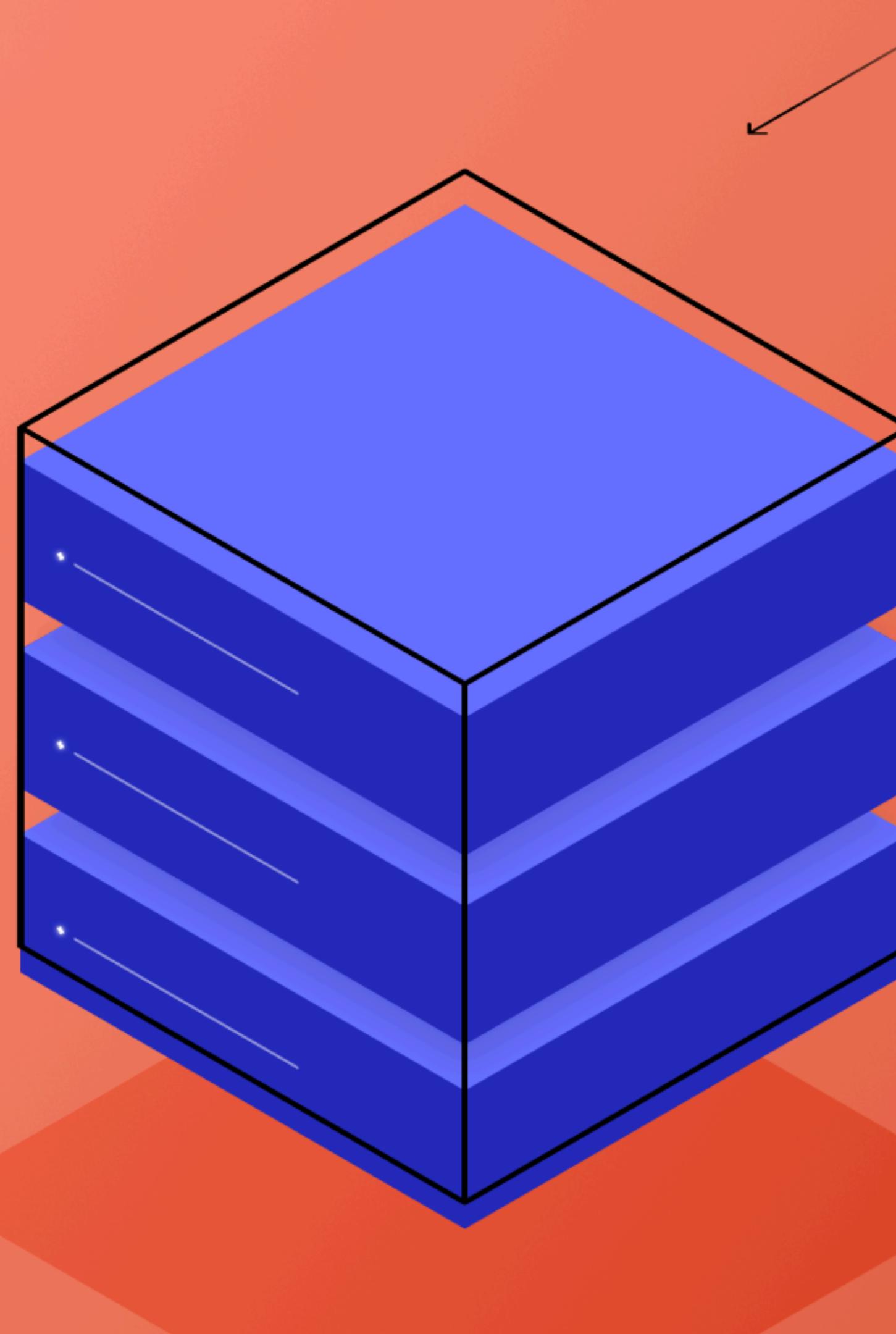
- Card
- ion-card
- ion-card-content
- ion-card-header
- ion-card-subtitle

Ionic Components

The screenshot shows the Ionic Components documentation for the `ion-accordion` component. The left sidebar lists various components: Getting Started, UI Components, Accordion (selected), Action Sheet, Alert, Badge, Breadcrumbs, Button, Card, and others. The main content area displays the `ion-accordion` component, which is described as providing collapsible sections. It includes a heading for "Basic Usage" and a code editor showing both the HTML template and TypeScript file. The right sidebar contains a "CONTENTS" menu with links to other accordion-related topics like Basic Usage, Toggle Accordions, Listen for Accordion State Changes, etc.

ionicframework.com/docs/components

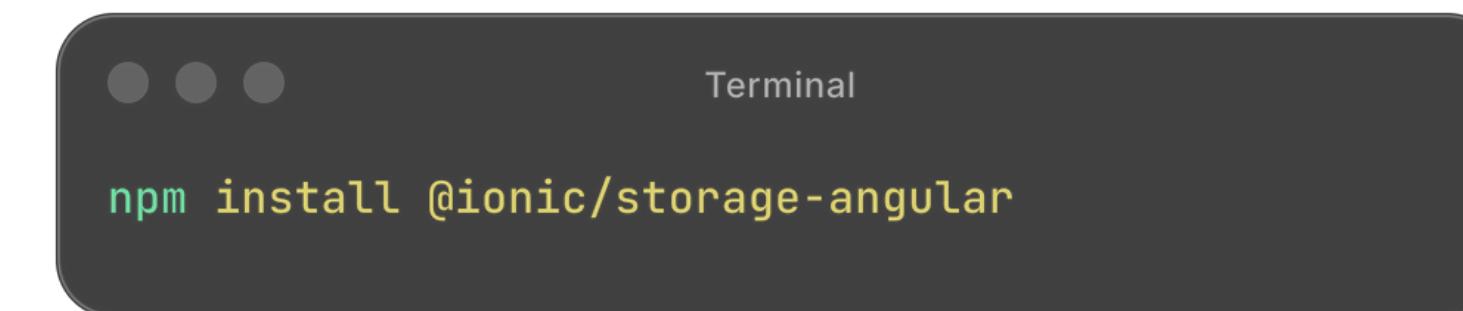
```
<ion-accordion>
  <ion-accordion value="first">
    <ion-item slot="header" color="light">
      <ion-label>First Accordion</ion-label>
    </ion-item>
```



Ionic Storage

O Ionic Storage é uma biblioteca para aplicações Ionic que oferece um sistema de armazenamento de dados simples, consistente e flexível.

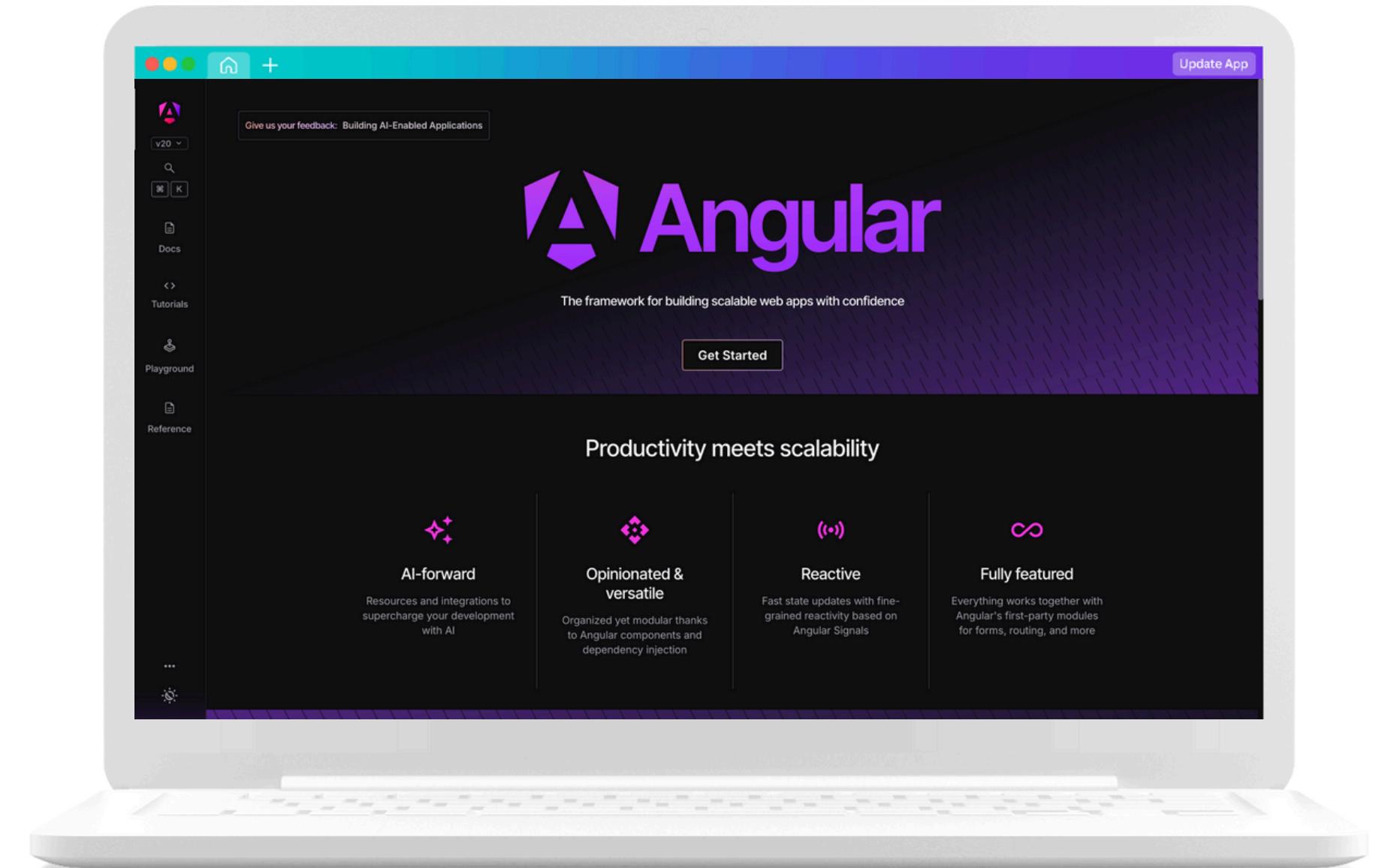
- Utiliza **IndexedDb**, **LocalStorage** ou **SQLite**
- Usa **Chave/Valor**
- Salva **objetos**
- **Compatibilidade entre frameworks**



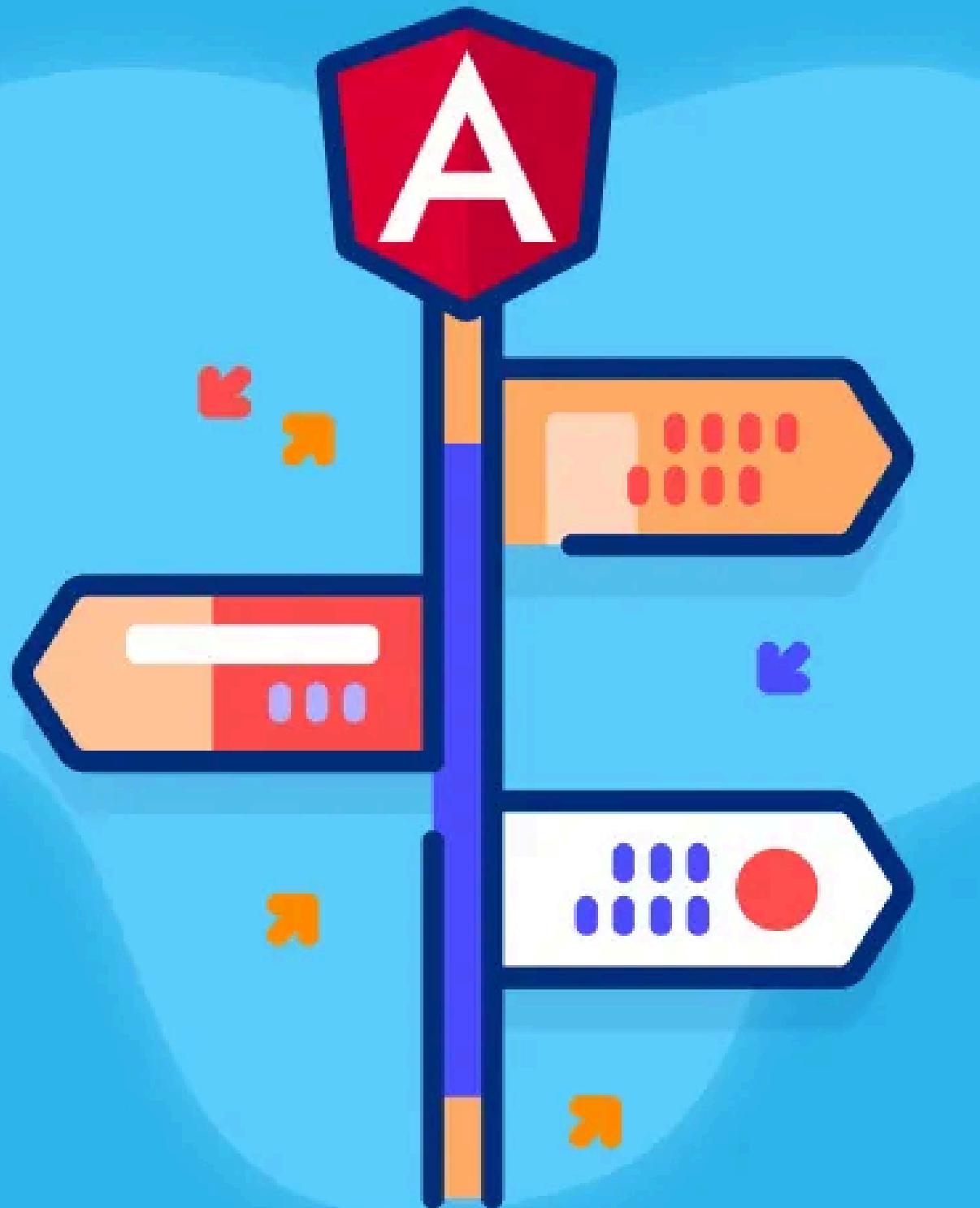
Angular

Single Page Application Framework

- Um framework de código aberto, desenvolvido pela Google
- Arquitetura baseada em componentes
- Utiliza **typescript**
- Principais características:
 - Two-way data binding
 - Roteamento
 - Modularidade
 - Injeção de dependência
 - Diretivas
 - Roteamento



Roteamento



- Mapea os componentes para rotas específicas
- Possibilita a navegação entre URLs sem recarregar a página
- Configuração e uso:
 - **RouterModule**
 - **RouterOutlet**
 - **RouterLink**
 - **ActivatedRoute**
- Além disso é possível configurar rotas filhas pra um componente

Diretivas no Angular

Estruturais

`*ngIf`

Altera a estrutura do DOM

Atributo

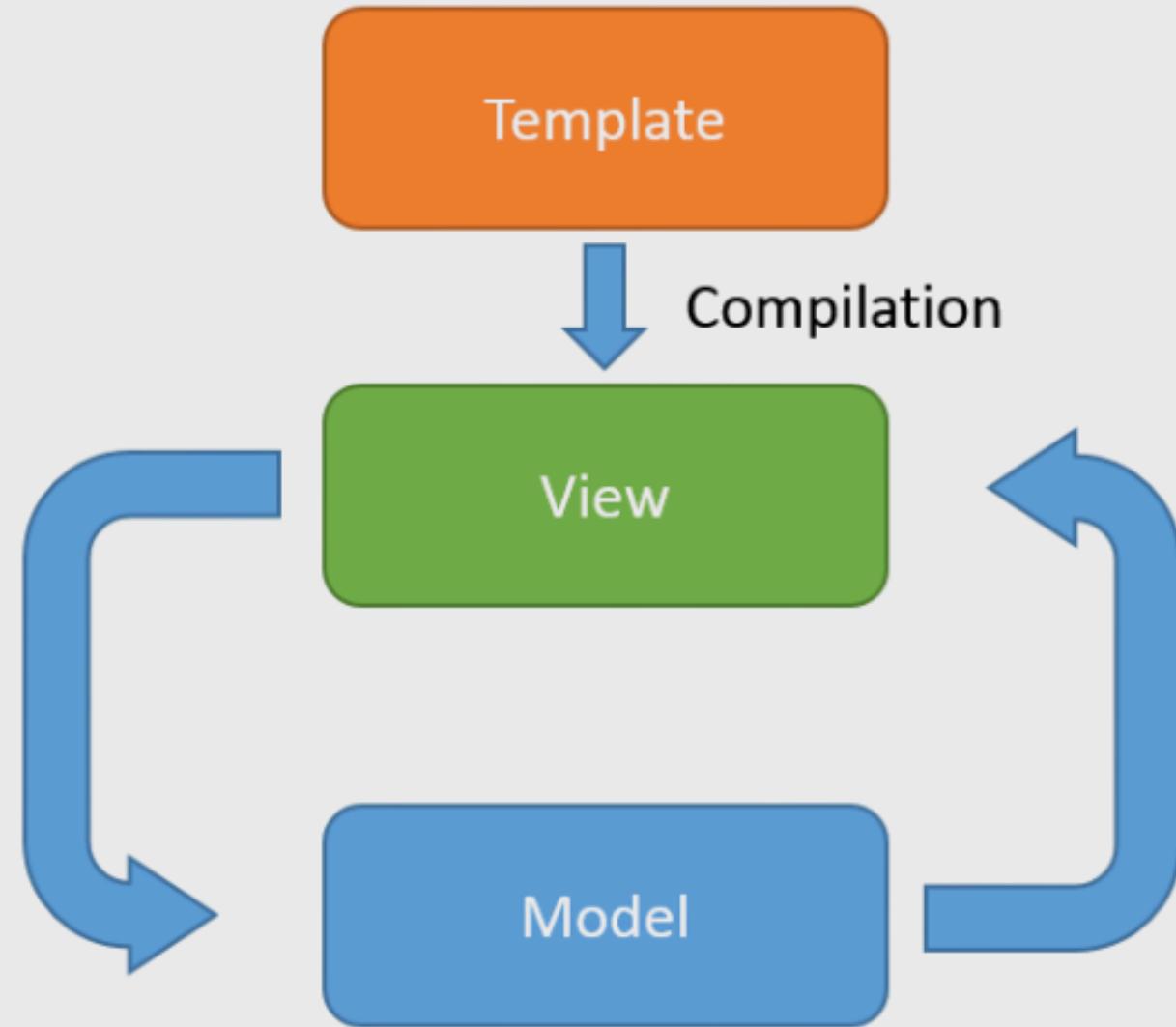
`[ngClass]`

Adiciona comportamento a um elemento

Personalizadas

`appHighlight`

Diretiva criada pelo usuário



Data Biding

- **Interpolação:** {{atributo}}
- **Property Binding:** [propriedade]="valor"
- **Event Binding:** (evento)="acao()"
- **Two-way Biding:** Template \rightleftarrows Componente

Forms

No Ionic com Angular, é possível utilizar os forms de duas formas principais:

- Template Driven
- Reactive Forms



Ferramentas e ambiente

Para executar em ambiente Web

- WebStorm ou VsCode (ou editor de preferência)
- Node
- npm
- Ionic CLI
- Git

Para executar em dispositivos móveis

- Ambiente com o Android ou IOS configurado
 - Android Studio pro Android
 - Xcode para IOS
 - Variáveis de ambiente
 - SDK do Android e Java
 - SDK do IOS (IOS 18)

Instalar o Node e npm

```
... Terminal  
npm install -g @ionic/cli
```

```
...  
ionic -v
```

Iniciando um novo projeto via terminal



Iniciar um novo projeto

```
ionic start app-to-do-minicurso blank --type=angular --capacitor
```



Executar o projeto localmente

```
ionic serve
```

Via AppFlow

ionic.io/appflow

Gerar novas páginas em um app

- 1.Criar o diretório pages
- 2.Ir ao diretório criado com o comando:



Terminal

```
cd src/app/pages
```

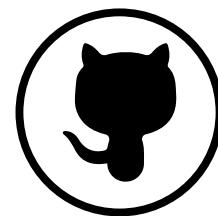
3. Gerar uma nova página:



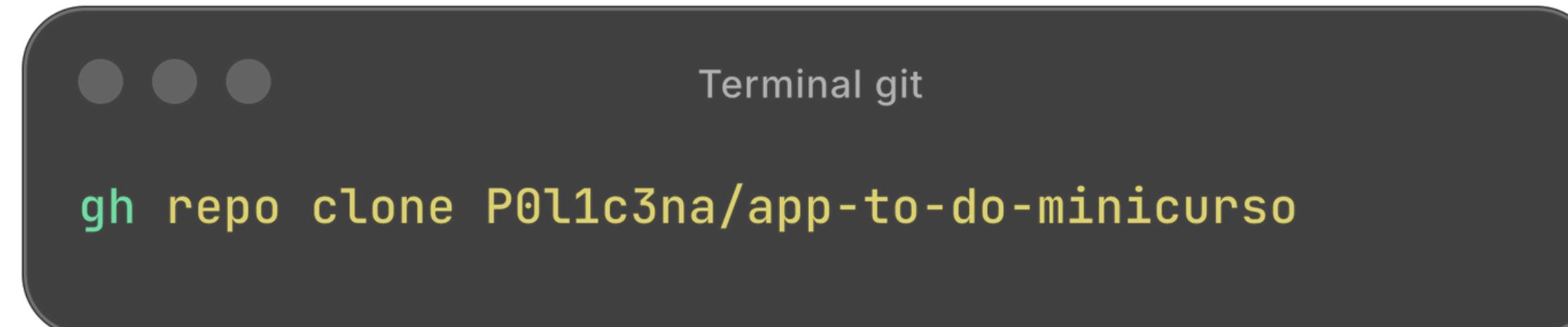
Terminal

```
ionic g page nome-da-pagina
```

Download do template do APP



github.com/P0l1c3na/app-to-do-minicurso



Adicionando o fab-button

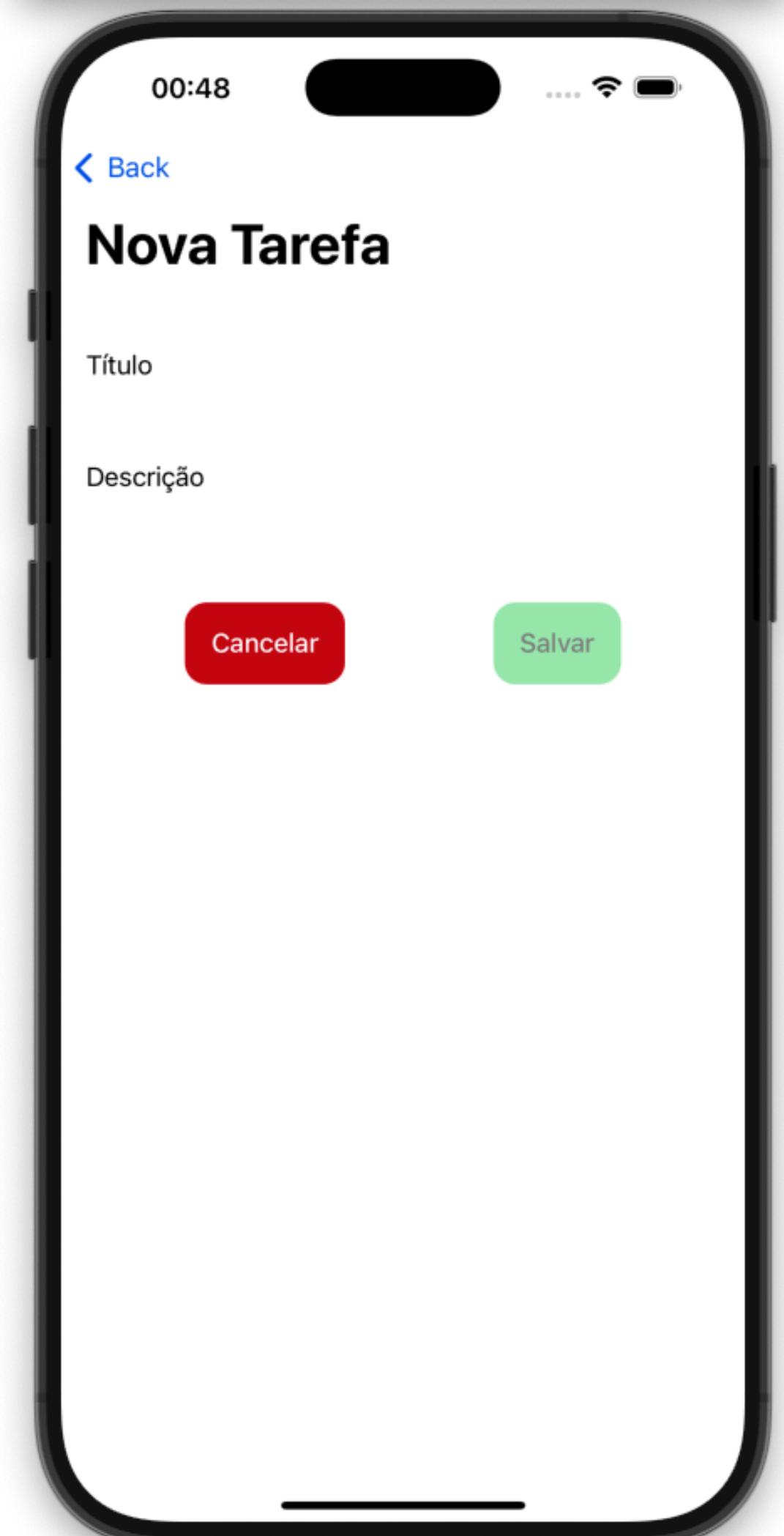
```
Template

<ion-fab class="ion-margin" slot="fixed" horizontal="end" vertical="bottom">
  <ion-fab-button routerLink="/nova-tarefa">
    <ion-icon name="add"></ion-icon>
  </ion-fab-button>
</ion-fab>
```



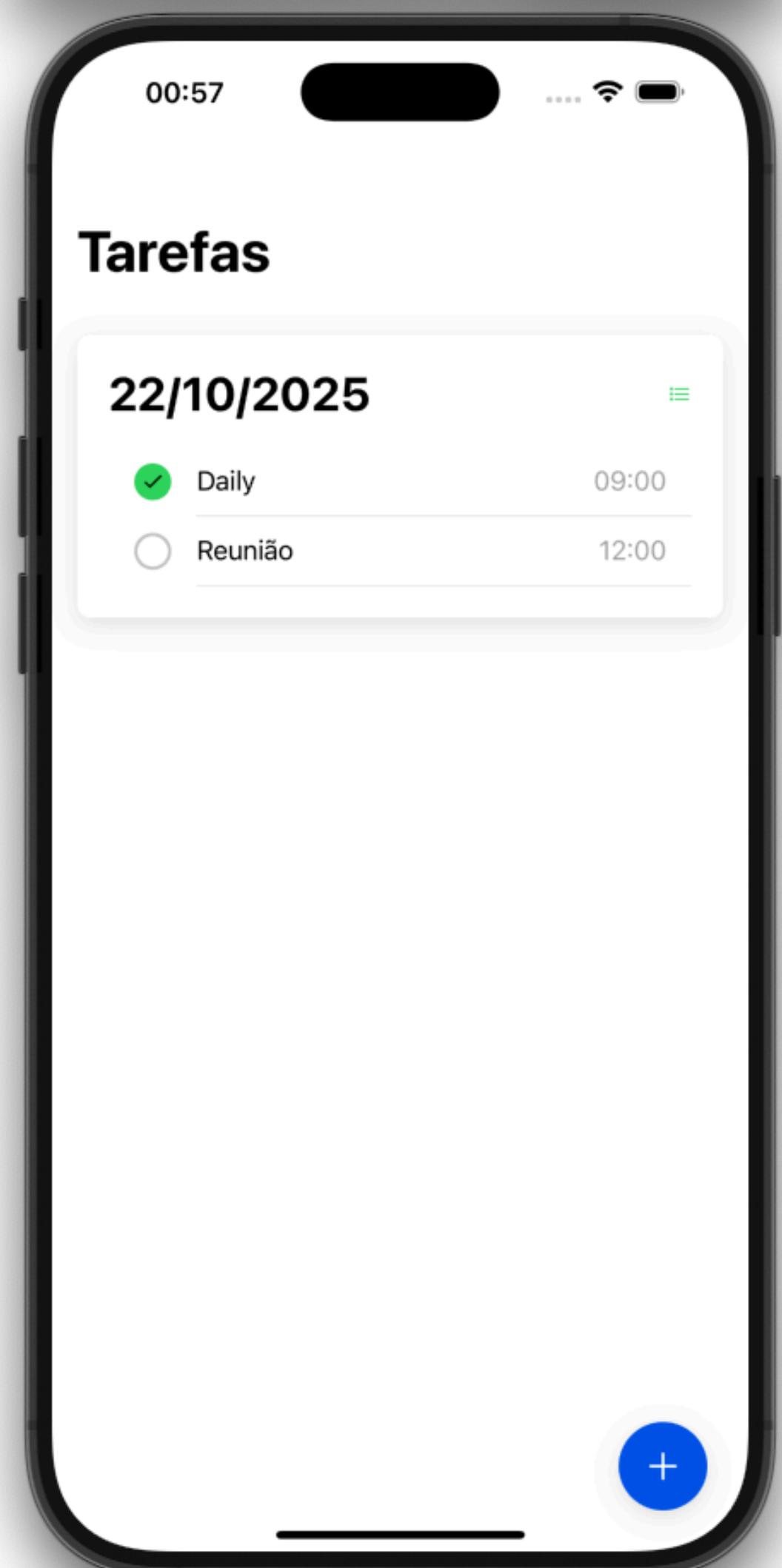
Tela de nova tarefa

- 1.Criar o model com o id, titulo, descrição e concluído
- 2.Criar o form para salvar e controlar os dados da tarefa
- 3.Adicionar os componentes de ion-input e ion-textarea
- 4.Implementar a service de tarefas para controlar e salvar as tarefas em memória
- 5.Criar a lógica do botão de Salvar e Cancelar



Tela de Tarefas

1. Obter a lista de tarefas cadastradas do tarefas service
2. Criar uma listagem utilizando ion-item, ion-checkbox e @for para mostrar as tarefas recuperadas e marca-las como concluídas ou pendentes



Melhorias na tela de Tarefas

1. Alterar o tarefas service para utilizar o Ionic Storage, a fim de persistir os dados
2. Implementar a funcionalidade de atualizar a tarefa ao marcar ou desmarcar um item
3. Adicionar o Ionic Toast (mensagens de feedback) ao marcar e desmarcar um item
4. Atualizar o Layout adicionando a quantidade de itens A Fazer e Feitos



Comandos de Build and Run para o Android

1. Construir o projeto



Terminal

```
ng build
```

2. Instalar e adicionar o plugin de build e deploy no android ao projeto



Terminal

```
npm install @capacitor/android
npx cap add android
```

3. Sincronizar os arquivos de build com o android



Terminal

```
cap sync
```

4. Abrir o projeto no Android Studio



Terminal

```
cap open android
```

Comandos de Build and Run para o IOS

1. Construir o projeto

```
Terminal  
ng build
```

2. Instalar e adicionar o plugin de build e deploy no IOS ao projeto

```
Terminal  
npm install @capacitor/ios  
npx cap add ios
```

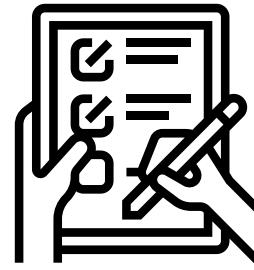
3. Sincronizar os arquivos de build com o android

```
Terminal  
cap sync
```

4. Abrir o projeto no Android Studio

```
Terminal  
cap open ios
```

Atividades



1. Adicionar horario e data separado (utilizar componente do Ionic ion-datetime)
2. Adicionar etiquetas
3. Alterar a tela "Nova tarefa" para abrir em um ion-modal e diretamente na tela de tarefas
4. Adicionar as tarefas concluídas em outra lista e apresentá-las separadamente
5. Possibilitar editar uma tarefa

Ionic + Angular

RAPHAEL POLICENA



phaelpolicena



raphael-policena



POl1c3na



phaelpolicena@gmail.com