**Revisão de Ionic e Angular**

**1. Estrutura do Projeto (Ionic + Angular)**

* src/app/: onde estão os componentes, páginas e módulos.
* app.module.ts: módulo principal do aplicativo.
* app-routing.module.ts: gerenciamento de rotas.
* assets/: arquivos estáticos (imagens, JSON etc).
* environments/: configurações para dev e produção.

**2. Conceitos Angular Essenciais**

* **Componentes (@Component)**
  + Template, CSS e lógica TS.
* **Módulos (@NgModule)**
  + Organização e importação de funcionalidades.
* **Serviços (@Injectable)**
  + Lógica reutilizável, injeção de dependência.
* **Diretivas**
  + Estruturais: \*ngIf, \*ngFor.

**3. Componentes e Páginas no Ionic**

* Gerar com CLI: ionic generate page nome ou ionic generate component nome

**4. Interface e Componentes Visuais (UI)**

* **Ionic Components**
  + Botões (ion-button)
  + Inputs (ion-input, ion-textarea)
  + Cards (ion-card)
  + Listas (ion-list, ion-item)
  + Modais, Alerts, Toasts, Loading
* **Temas e Estilização**
  + Utilização de CSS Variables
  + Customização de cores em variables.scss

**5. Forms (Formulários)**

* Template-driven Forms (FormsModule)
* Reactive Forms (ReactiveFormsModule)
* Validações com Validators
* Mostrar mensagens de erro com \*ngIf

**6. Comunicação com API (HTTPClient)**

* Importação de HttpClientModule
* Métodos: get, post, put, delete
* Uso de subscribe() para consumir respostas
* Boas práticas com services para requisições

**7. Serviços e Injeção de Dependência**

* Criar com CLI: ionic generate service nome
* Compartilhamento de dados entre componentes/páginas

**8. Rotas e Navegação**

* Configuração no app-routing.module.ts

**9. Armazenamento Local**

* **Ionic Storage** (armazenamento assíncrono)
  + Instalar: npm install @ionic/storage-angular

**10. Capacitor (Integração com nativo)**

* Instalar plugins: npx cap add android, npx cap add ios
* Comandos:
  + npx cap sync
  + npx cap open android

**11. Build e Deploy**

* **Desenvolvimento**
  + ionic serve (modo web)
* **Produção**
  + ionic build --prod
  + npx cap copy + npx cap open android

**12. Boas Práticas**

* Reutilizar componentes e serviços
* Usar interfaces para tipos (.ts)