

# Resumo Programação Mobile Coding

## UNIDADE III – Interface Gráfica e Layout no Android (Flutter)

### Conceitos Básicos de Layout com Flutter

O Flutter usa **widgets** para construir interfaces. Existem dois tipos principais:

#### ♦ *Widgets de Layout*

- **Row:** Alinha elementos **horizontalmente**.  
*Exemplo:* Ícones lado a lado em uma barra.
- **Column:** Alinha elementos **verticalmente**.  
*Exemplo:* Título, conteúdo e rodapé.
- **Stack:** Sobre põe elementos.  
*Exemplo:* Ícone em cima de uma imagem.

#### ♦ *Widgets de Visualização*

- **Text:** Exibe textos personalizáveis (cor, fonte, tamanho).
- **Image:** Mostra imagens estáticas ou dinâmicas (PNG, JPEG).
- **Icon:** Mostra ícones vetoriais (ex: ícone de câmera).

### Customização de Estilos

- **ThemeData** e **TextStyle** são usados para aplicar cores, fontes e ícones de forma **consistente** em toda a interface.

### Geradores de Layout

- **Expanded, Flexible, Spacer:** Ajudam a controlar o espaço entre elementos dentro de uma Row/Column.

## Posicionamento e Alinhamento

- Pode-se alinhar widgets em qualquer lugar com precisão usando:
  - `mainAxisAlignment`, `crossAxisAlignment`
  - `Transform`, `Align`

## Layouts Responsivos

- **Responsividade:** Adapta o layout a diferentes tamanhos de tela (celular, tablet).
  - **Breakpoints:** Mudam o layout conforme a largura da tela.
  - **Flexibilidade:** Ajusta conteúdo dinâmico sem quebrar o design.
- 

## UNIDADE IV – Banco de Dados Móveis

### O Que São Bancos de Dados Móveis?

São bancos projetados para rodar **localmente em dispositivos móveis**:

- **Mobilidade:** Acesso em qualquer lugar.
- **Offline:** Funciona sem internet e sincroniza depois.
- **Otimização:** Usa poucos recursos (memória, bateria).

### SQLite – Banco Embarcado

- **Mais usado no mundo.**
- **Leve e sem necessidade de servidor.**

- **Biblioteca local** que roda junto com o app.

#### ♦ Recursos do SQLite:

- Não precisa configurar servidor.
- Permite **CRUD** (Create, Read, Update, Delete):
  - `insert()`, `update()`, `delete()`, `query()`, `execSQL()`
- Armazena os dados localmente no caminho `getDatabasesPath()`.

#### ✓ Vantagens:

- **Rápido**, leve, fácil de usar com Flutter.
- Armazena dados de forma **persistente**, mesmo com o app fechado.

#### 🔧 Exemplo Didático:

dart

```
// Inserir dados com Flutter + SQLite
await db.insert(
  'usuarios',
  {'nome': 'Paulo', 'idade': 18},
);
```

---

## 🌐 UNIDADE IV – Aplicações Híbridas Mobile

### 📌 O Que São Apps Híbridos?

- Combinam **tecnologias web** (HTML, CSS, JS) com um **container nativo**.
- Executam o app em **diferentes plataformas (Android/iOS)** com um **único código**.

#### ⚙️ WebView

- É o "coração" do app híbrido.
- Renderiza conteúdo web dentro do app.

### Comparação Entre Tipos de Aplicativos:

Tipo	Prós	Contras
<b>Nativo</b>	Melhor desempenho, acesso total ao hardware	Código duplicado p/ cada plataforma
<b>Web App</b>	Acessível via navegador	Acesso limitado ao dispositivo
<b>Híbrido</b>	Código único, mais rápido e barato	Desempenho inferior em tarefas pesadas

### Vantagens de Apps Híbridos:

- Economia de tempo e custo.
- Atualizações rápidas.
- Ideal para **equipes pequenas**.

### Limitações:

- Desempenho menor em tarefas pesadas (ex: jogos 3D).
- Acesso limitado a sensores e recursos do celular.

### Exemplos Reais:

- **Uber** (usou PhoneGap nas versões iniciais)
- **Instagram/Twitter** (usam WebView para partes do app)