Desenvolvimento para Dispositivos Móveis

Aula 05

Layouts e componentes básicos de interface

Prof. Msc. Francisco de Assis de Souza Rodrigues

UniFAP
CENTRO UNIVERSITÁRIO PARAÍSO

Visão Geral

- Introdução ao Design de Interfaces para Dispositivos Móveis
- Tipos de Layouts em Desenvolvimento Mobile
- Componentes Básicos de Interface
- Atividade
- Desafio



UniFAP
CENTRO UNIVERSITÁRIO PARAÍSO

Introdução ao Design de Interfaces para Dispositivos Móveis

O design de interfaces para dispositivos móveis (Mobile UI Design) é uma área essencial no desenvolvimento de aplicativos, pois influencia diretamente na usabilidade e na experiência do usuário (UX). Um bom design pode tornar um aplicativo intuitivo, agradável e eficiente, enquanto um design ruim pode afastar os usuários.

UniFAP
CENTRO UNIVERSITÁRIO PARAÍSO

Introdução ao Design de Interfaces para Dispositivos Móveis

📌 O que são layouts e por que são importantes?

O **layout** define a organização dos elementos na tela, incluindo botões, imagens, textos e outros componentes interativos.

- Um layout **bem estruturado** melhora a usabilidade e a experiência do usuário.
- Em dispositivos móveis, o espaço da tela é limitado, então a organização deve ser eficiente e intuitiva.
- O design precisa se **adaptar a diferentes tamanhos de tela e resoluções**.

🔹 **Exemplo:** Um layout mal projetado pode dificultar a navegação e prejudicar a acessibilidade, tornando um app frustrante para o usuário.

UniFAP
CENTRO UNIVERSITÁRIO PARAÍSO

Introdução ao Design de Interfaces para Dispositivos Móveis

🎨 Diferença entre UI e UX

✅ UI (User Interface) - Interface do Usuário

Refere-se ao design visual e à disposição dos elementos gráficos do app, como botões, cores, tipografia e ícones.

✅ UX (User Experience) - Experiência do Usuário

Diz respeito a como o usuário interage com o aplicativo. Envolve facilidade de uso, eficiência e satisfação.

- **Exemplo:**

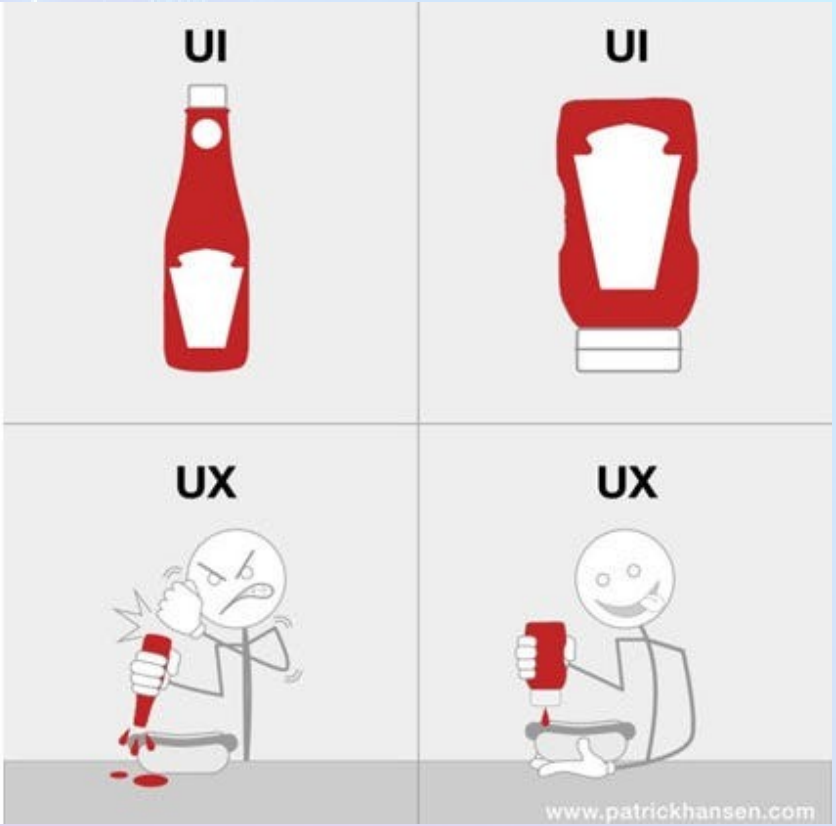
- **Boa UI, má UX:** Um app bonito, mas difícil de usar.
- **Boa UX, má UI:** Um app funcional, mas com um design pouco atrativo.
- **Boa UI e UX:** Um app visualmente agradável e fácil de usar.

UniFAP
CENTRO UNIVERSITÁRIO PARAÍSO

Introdução ao Design de Interfaces para Dispositivos Móveis

🎨 Diferença entre UI e UX





UniFAP
CENTRO UNIVERSITÁRIO PARAÍSO

Introdução ao Design de Interfaces para Dispositivos Móveis

📱 Padrões de Design para Mobile

Empresas como **Google e Apple** definiram diretrizes para criar interfaces consistentes e intuitivas:

1. **Material Design (Google - Android)**
    - Usa **cartões, sombras e animações** para criar interfaces modernas.
    - Foca na clareza e na hierarquia dos elementos.
    - Exemplo: Botão **Floating Action Button (FAB)** para ações principais.
2. **Human Interface Guidelines (Apple - iOS)**
    - Prioriza a simplicidade e a elegância no design.
    - Usa gestos naturais para navegação.
    - Exemplo: Uso de **transparências e animações suaves**.

📌 **Curiosidade:** O Material Design é mais padronizado, enquanto o iOS aposta em um design mais minimalista.

UniFAP
CENTRO UNIVERSITÁRIO PARAÍSO

Introdução ao Design de Interfaces para Dispositivos Móveis

🎯 Boas Práticas no Design de Interfaces Mobile

Simplicidade: Interfaces devem ser intuitivas, sem elementos desnecessários.

Tamanho dos elementos: Botões e textos devem ser legíveis e fáceis de tocar.

Consistência: Manter um padrão visual em todas as telas do app.

Acessibilidade: Cores contrastantes, fontes legíveis e suporte a leitores de tela.

UniFAP
CENTRO UNIVERSITÁRIO PARAÍSO

Introdução ao Design de Interfaces para Dispositivos Móveis

👨‍💻 Atividade Prática

📌 Análise um aplicativo que usam diariamente e responda:

1. O design da interface é intuitivo?
2. O app segue padrões como Material Design ou Human Interface Guidelines?
3. Há algo que poderia ser melhorado na experiência do usuário?

UniFAP
CENTRO UNIVERSITÁRIO PARAÍSO

Tipos de Layouts em Desenvolvimento Mobile

Os layouts em aplicativos móveis são responsáveis por organizar os elementos da interface na tela, garantindo uma boa experiência para o usuário. Dependendo do sistema operacional e da tecnologia utilizada, existem diferentes tipos de layouts.

- **LinearLayout**
- **RelativeLayout / Constraint Layout**
- **FrameLayout**
- **GridLayout**
- **ScrollView**

Android utiliza XML e pode ser manipulado programaticamente no código.

Tipos de Layouts em Desenvolvimento Mobile

LinearLayout

📌 **Organiza os elementos em uma única direção (vertical ou horizontal).**

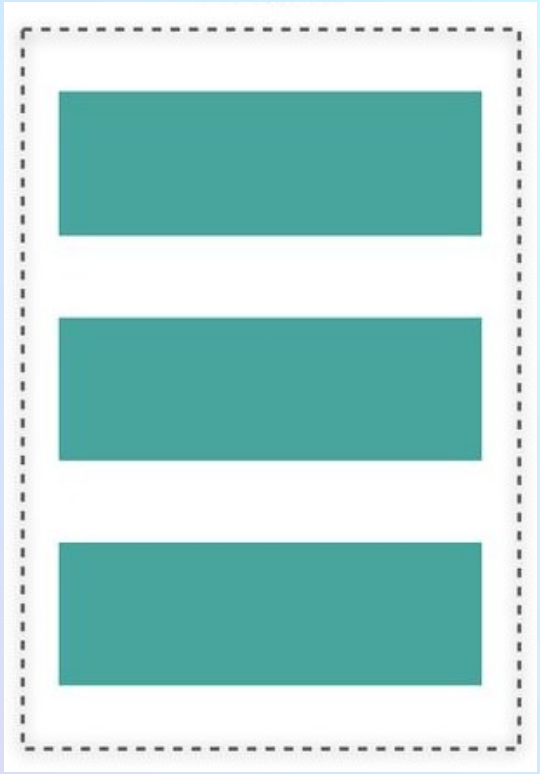
- Os componentes são dispostos em sequência.
- Usa o atributo android:orientation="vertical" ou horizontal.
- Simples, mas pode gerar aninhamento excessivo (não recomendado para layouts complexos).

UniFAP
CENTRO UNIVERSITÁRIO PARAÍSO

```xml
<LinearLayout
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <TextView
        android:text="Bem-vindo!"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"/>

    <Button
        android:text="Clique aqui"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"/>
</LinearLayout>
```



https://developer.android.com/develop/ui/views/layout/linear?hl=pt-br

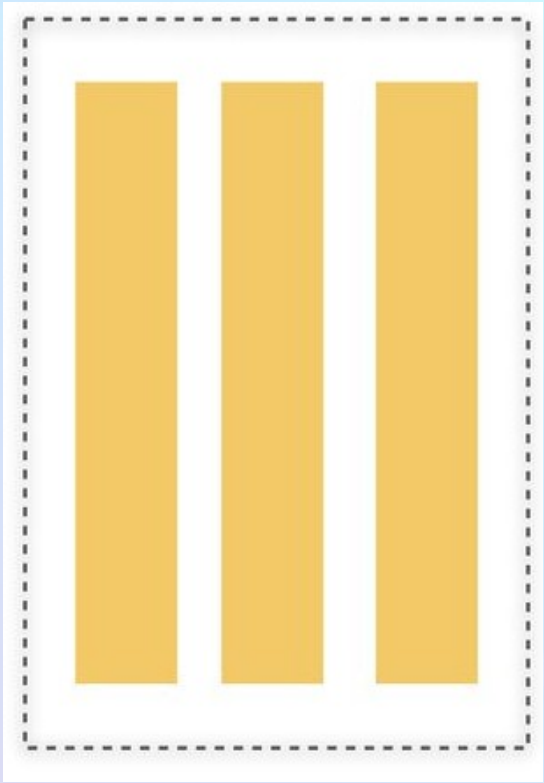
```xml
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:padding="16dp">

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Botão 1"
        android:layout_marginEnd="16dp"/>

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Botão 2"
        android:layout_marginEnd="16dp"/>

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Botão 3"/>

</LinearLayout>
```



LinearLayout

UniFAP
CENTRO UNIVERSITÁRIO PARAÍSO

Tipos de Layouts em Desenvolvimento Mobile

ConstraintLayout (Recomendado)

📌 **Permite posicionar elementos de forma flexível, sem precisar aninhar muitos layouts.**

- Reduz a complexidade e melhora o desempenho.
- Os elementos são posicionados com base em **restrições** (constraints).
- Substitui o RelativeLayout e é o padrão recomendado atualmente.

🔹 **Vantagens:** Mais eficiente, evita aninhamento desnecessário.

UniFAP
CENTRO UNIVERSITÁRIO PARAÍSO

Tipos de Layouts em Desenvolvimento Mobile

ConstraintLayout (Recomendado)

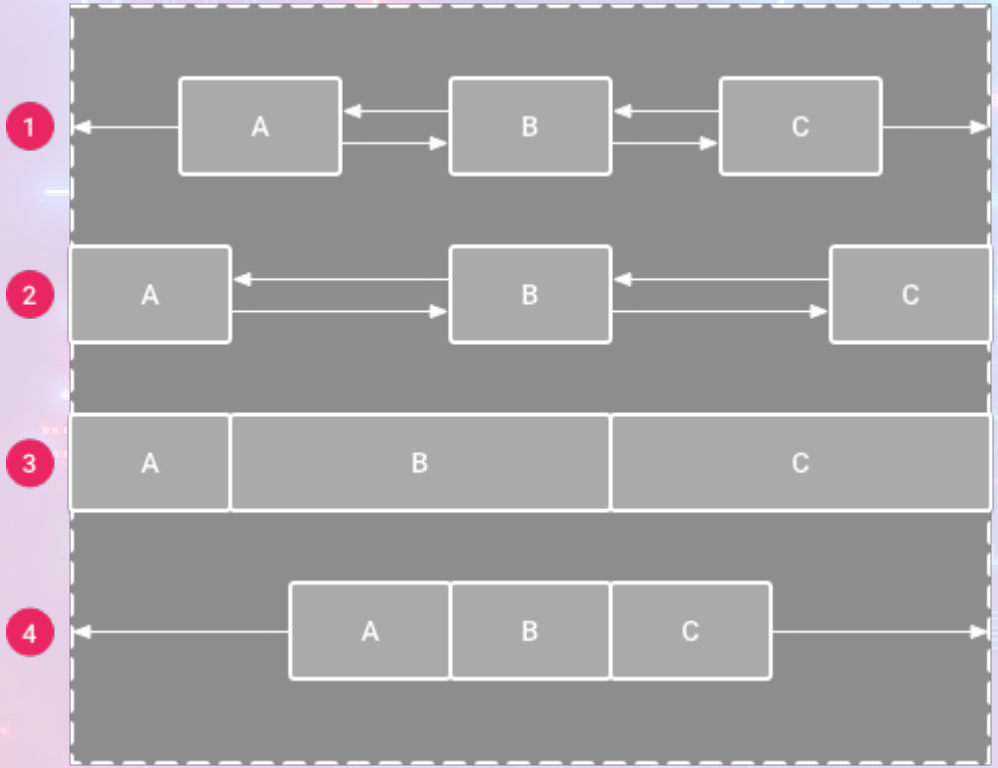
```
<androidx.constraintlayout.widget.ConstraintLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <Button
        android:text="Botão"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintEnd_toEndOf="parent"/>
</androidx.constraintlayout.widget.ConstraintLayout>
```

UniFAP
CENTRO UNIVERSITÁRIO PARAÍSO

Tipos de Layouts em Desenvolvimento Mobile

ConstraintLayout (Recomendado)



https://developer.android.com/develop/ui/views/layout/constraint-layout?hl=pt-br

UniFAP
CENTRO UNIVERSITÁRIO PARAÍSO

Tipos de Layouts em Desenvolvimento Mobile

RelativeLayout

📌 **Os elementos são posicionados de forma relativa a outros componentes.**

- Permite organizar componentes com android:layout_below, layout_alignParentStart, etc.
- Era muito usado antes da introdução do ConstraintLayout.

🔹 **Desvantagem:** Pode se tornar complexo e de difícil manutenção.

UniFAP
CENTRO UNIVERSITÁRIO PARAÍSO

Tipos de Layouts em Desenvolvimento Mobile

RelativeLayout

```
<RelativeLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <TextView
        android:id="@+id/titulo"
        android:text="Título"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true"/>

    <Button
        android:text="Clique aqui"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@id/titulo"
        android:layout_centerHorizontal="true"/>
</RelativeLayout>
```



https://developer.android.com/develop/ui/views/layout/relative?hl=pt-br

Tipos de Layouts em Desenvolvimento Mobile

FrameLayout

📌 **Usado para sobrepor elementos, ideal para telas de carregamento ou efeitos visuais.**

- Permite empilhar elementos uns sobre os outros.
- O primeiro elemento adicionado é a base, os outros aparecem por cima.

🔹 **Usado em:** Telas de carregamento, sobreposição de elementos.

UniFAP
CENTRO UNIVERSITÁRIO PARAÍSO

Tipos de Layouts em Desenvolvimento Mobile

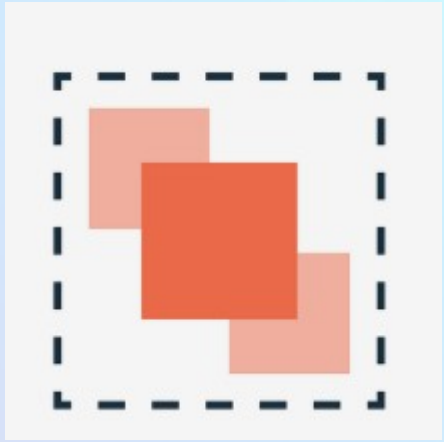
FrameLayout

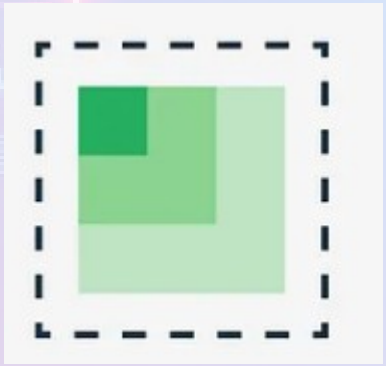
```
<FrameLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <ImageView
        android:src="@drawable/background"
        android:layout_width="match_parent"
        android:layout_height="match_parent"/>

    <ProgressBar
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"/>
</FrameLayout>
```

https://developer.android.com/reference/android/widget/FrameLayout





UniFAP
CENTRO UNIVERSITÁRIO PARAÍSO

Tipos de Layouts em Desenvolvimento Mobile

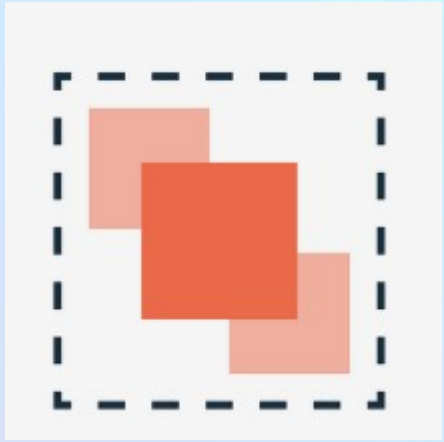
FrameLayout

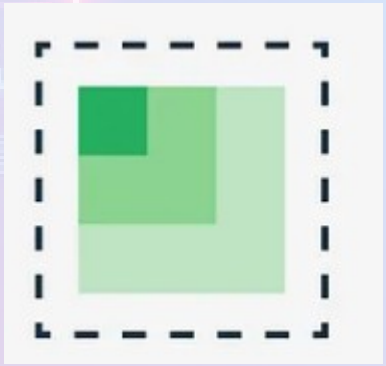
```
<FrameLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <ImageView
        android:src="@drawable/background"
        android:layout_width="match_parent"
        android:layout_height="match_parent"/>

    <ProgressBar
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"/>
</FrameLayout>
```

https://developer.android.com/reference/android/widget/FrameLayout





UniFAP
CENTRO UNIVERSITÁRIO PARAÍSO

Tipos de Layouts em Desenvolvimento Mobile

GridLayout

📌 **Organiza os elementos em uma grade, semelhante a uma tabela.**

- Permite organizar elementos em **linhas e colunas**.
- Útil para exibir **galerias de imagens, teclados e layouts tabulares**.

🔹 **Usado para:** Criar layouts organizados em colunas, como menus ou galerias.

UniFAP
CENTRO UNIVERSITÁRIO PARAÍSO

Tipos de Layouts em Desenvolvimento Mobile

GridLayout

```
<GridLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:columnCount="2">

    <Button
        android:text="Botão 1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_columnSpan="1"/>

    <Button
        android:text="Botão 2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_columnSpan="1"/>
</GridLayout>
```



UniFAP
CENTRO UNIVERSITÁRIO PARAÍSO

Tipos de Layouts em Desenvolvimento Mobile

ScrollView

📌 **Permite que o usuário role a tela quando há mais conteúdo do que cabe na tela.**

- Pode ser usado com um LinearLayout dentro.

- 🔹 **Usado quando:** O conteúdo da tela pode ser maior que a tela do dispositivo.

UniFAP
CENTRO UNIVERSITÁRIO PARAÍSO

Tipos de Layouts em Desenvolvimento Mobile

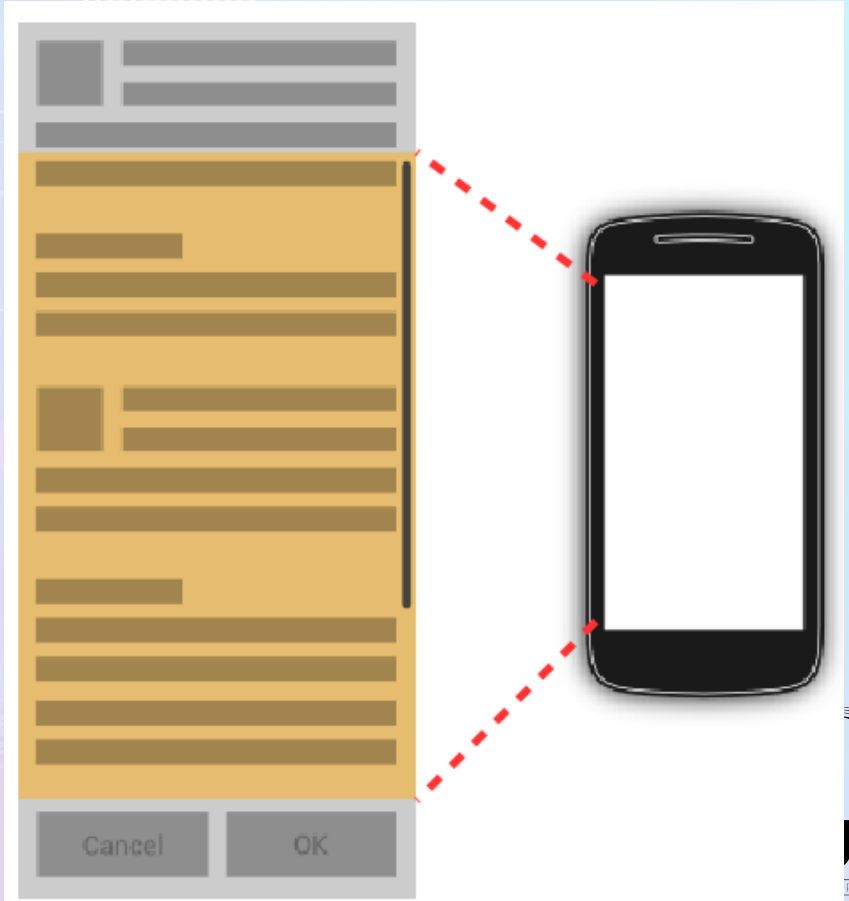
ScrollView

```
<ScrollView
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <LinearLayout
        android:orientation="vertical"
        android:layout_width="match_parent"
        android:layout_height="wrap_content">

        <TextView
            android:text="Texto longo..."
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"/>

        <Button
            android:text="Botão"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"/>
    </LinearLayout>
</ScrollView>
```



Tipos de Layouts em Desenvolvimento Mobile

Resumo: Qual Layout Escolher?

Tipo de Layout	Características	Quando Usar?
LinearLayout	Simples, sequencial	Layouts pequenos
ConstraintLayout	Flexível e eficiente	Layouts modernos e complexos
FrameLayout	Sobreposição de elementos	Telas de carregamento, efeitos visuais
RelativeLayout	Organização baseada em relação entre elementos	Layouts antigos (substituído pelo ConstraintLayout)
GridLayout	Organiza elementos em grade	Galerias, tabelas, menus visuais
ScrollView	Adiciona rolagem	Conteúdos longos

UniFAP
CENTRO UNIVERSITÁRIO PARAÍSO

Atividade

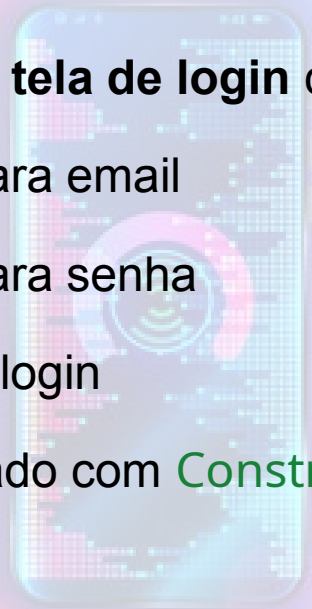
Criar um layout de **tela de login** com:

✅ Um EditText para email

✅ Um EditText para senha

✅ Um Button de login

✅ Tudo centralizado com ConstraintLayout



UniFAP
CENTRO UNIVERSITÁRIO PARAÍSO

Desafio

- Criar uma **tela de perfil** usando LinearLayout ou ConstraintLayout.



UniFAP
CENTRO UNIVERSITÁRIO PARAÍSO

Componentes Básicos de Interface

Principais widgets/componentes gráficos usados para interação com o usuário:

- **TextView / Label:** Exibição de texto.
- **EditText / TextField:** Entrada de texto pelo usuário.
- **Button:** Botões para ações.
- **ImageView:** Exibição de imagens.
- **RecyclerView / ListView:** Listas de elementos dinâmicos.
- **Switch, Checkbox, RadioButton:** Opções de seleção.

UniFAP
CENTRO UNIVERSITÁRIO PARAÍSO

Componentes Básicos de Interface

TextView / Label

Função: Exibe texto na tela.

Uso: Geralmente usado para mostrar informações estáticas, como rótulos de campos, títulos, ou mensagens.

```
<TextView
    android:id="@+id/textView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Bem-vindo ao NExTI!" />
```

UniFAP
CENTRO UNIVERSITÁRIO PARAÍSO

Componentes Básicos de Interface

EditText / TextField

Função: Permite que o usuário insira texto.

Uso: Usado em formulários para coletar informações como nome, e-mail, ou qualquer dado que envolva texto.

```
<EditText
    android:id="@+id/editText"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="Digite seu nome" />
```

UniFAP
CENTRO UNIVERSITÁRIO PARAÍSO

Componentes Básicos de Interface

Button

Função: Um botão que o usuário pode clicar para realizar uma ação, como enviar um formulário ou iniciar um processo.

Uso: Essencial para qualquer ação no app, como enviar dados ou navegar entre telas.

```
<Button
    android:id="@+id/button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Enviar"
    android:onClick="onClickButton" />
```

UniFAP
CENTRO UNIVERSITÁRIO PARAÍSO

Componentes Básicos de Interface]

ImageView

Função: Exibe imagens.

Uso: Ideal para mostrar imagens estáticas ou dinâmicas, como ícones, fotos, ou outros gráficos.

```
<ImageView
    android:id="@+id/imageView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:src="@drawable/icone_app" />
```

UniFAP
CENTRO UNIVERSITÁRIO PARAÍSO

Componentes Básicos de Interface

RecyclerView / ListView

Função: Exibe uma lista de itens em um layout de rolagem, permitindo desempenho eficiente para grandes conjuntos de dados.

Uso: Usado em casos em que se deseja exibir uma lista ou grades de itens, com possibilidade de personalização do item da lista.

```
<androidx.recyclerview.widget.RecyclerView
    android:id="@+id/recyclerView"
    android:layout_width="match_parent"
    android:layout_height="match_parent" />
```

UniFAP
CENTRO UNIVERSITÁRIO PARAÍSO

Componentes Básicos de Interface

Switch, Checkbox, RadioButton

Função: Componentes para selecionar opções ou alterar estados.

- **Switch**: Para ativar ou desativar uma opção.
- **Checkbox**: Permite que o usuário marque múltiplas opções em uma lista.
- **RadioButton**: Permite que o usuário selecione apenas uma opção de um conjunto de opções.

Uso:

- **Switch**: Usado quando a escolha é binária (ativar/desativar).
- **Checkbox**: Usado quando várias opções podem ser selecionadas.
- **RadioButton**: Usado para um único item em um conjunto de opções mutuamente exclusivas.

UniFAP
CENTRO UNIVERSITÁRIO PARAÍSO

Componentes Básicos de Interface

Switch, Checkbox, RadioButton

UniFAP
CENTRO UNIVERSITÁRIO PARAÍSO

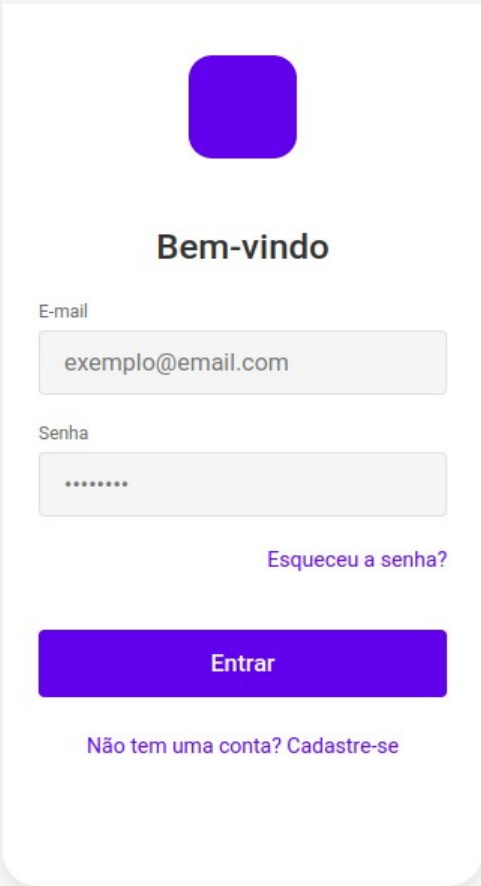
```
<!-- Switch -->
<Switch
    android:id="@+id/switch"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Ativar notificação" />

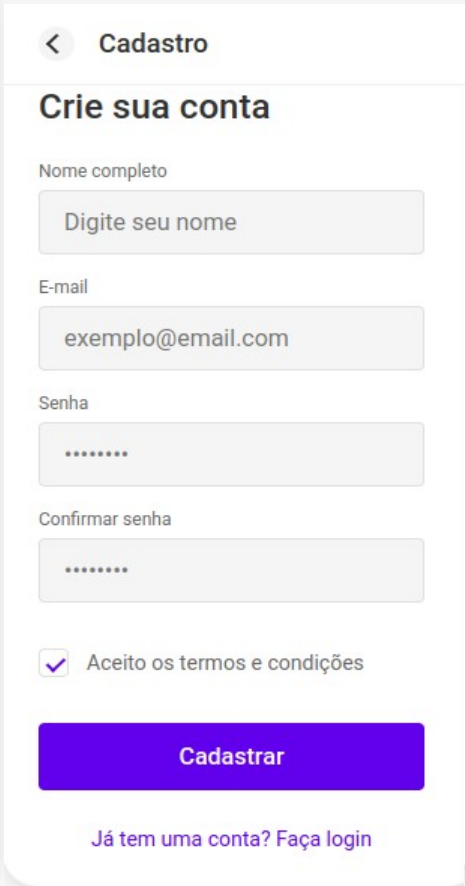
<!-- Checkbox -->
<CheckBox
    android:id="@+id/checkBox"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Aceito os termos" />
```

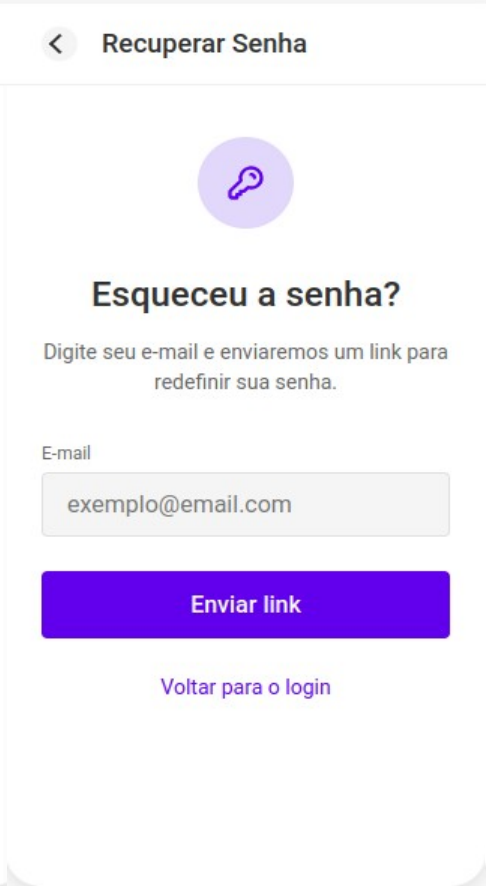
```
<!-- RadioButton -->
<RadioGroup>
    <RadioButton
        android:id="@+id/radioButton1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Opção 1" />
    <RadioButton
        android:id="@+id/radioButton2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Opção 2" />
</RadioGroup>
```

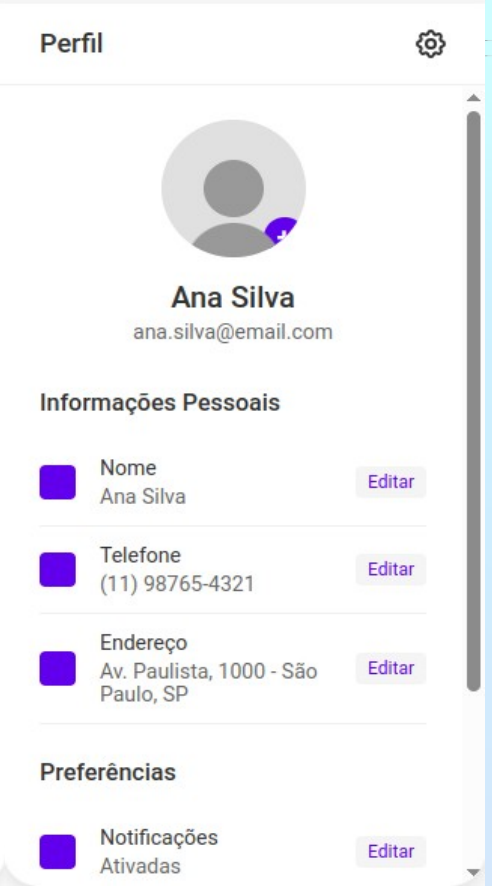
Atividade

Crie as interfaces gráficas a seguir:









Atividade

Implemente as funcionalidades

- **Login**: o usuário deve inserir email e senha, verifique se está correto e retorne uma notificação Toast se login realizado correto ou não.
- **Cadastro:** crie a lógica para cadastro de usuário e retorne se foi efetuado com sucesso ou não, não é necessário registrar os dados de fato, apenas responder ao usuário via Toast.

Bibliografia

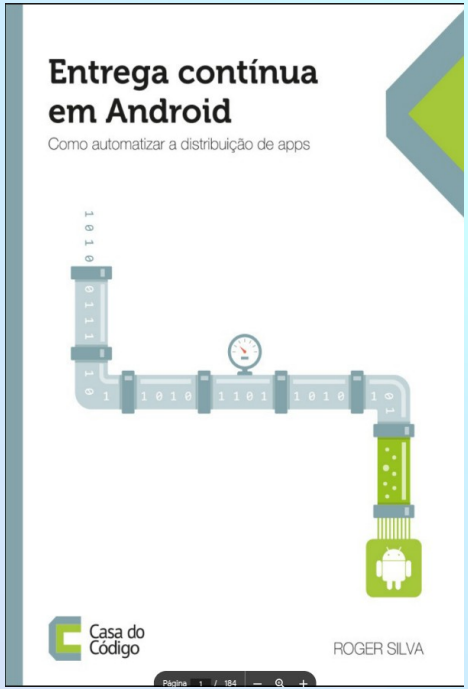
RESENDE, Kassiano. Kotlin com android: crie aplicativos de maneira fácil e divertida. São Paulo, SP: Casa do Código, 2018. E-book. Disponível em: https://plataforma.bvirtual.com.br. Acesso em: 26 fev. 2025.

https://plataforma.bvirtual.com.br/Acervo/Publicacao/213016

SILVA, Roger. Entrega contínua em android: como automatizar a distribuição de apps. São Paulo, SP: Casa do Código, 2016. E-book. Disponível em: https://plataforma.bvirtual.com.br. Acesso em: 26 fev. 2025.

https://plataforma.bvirtual.com.br/Acervo/Publicacao/212670

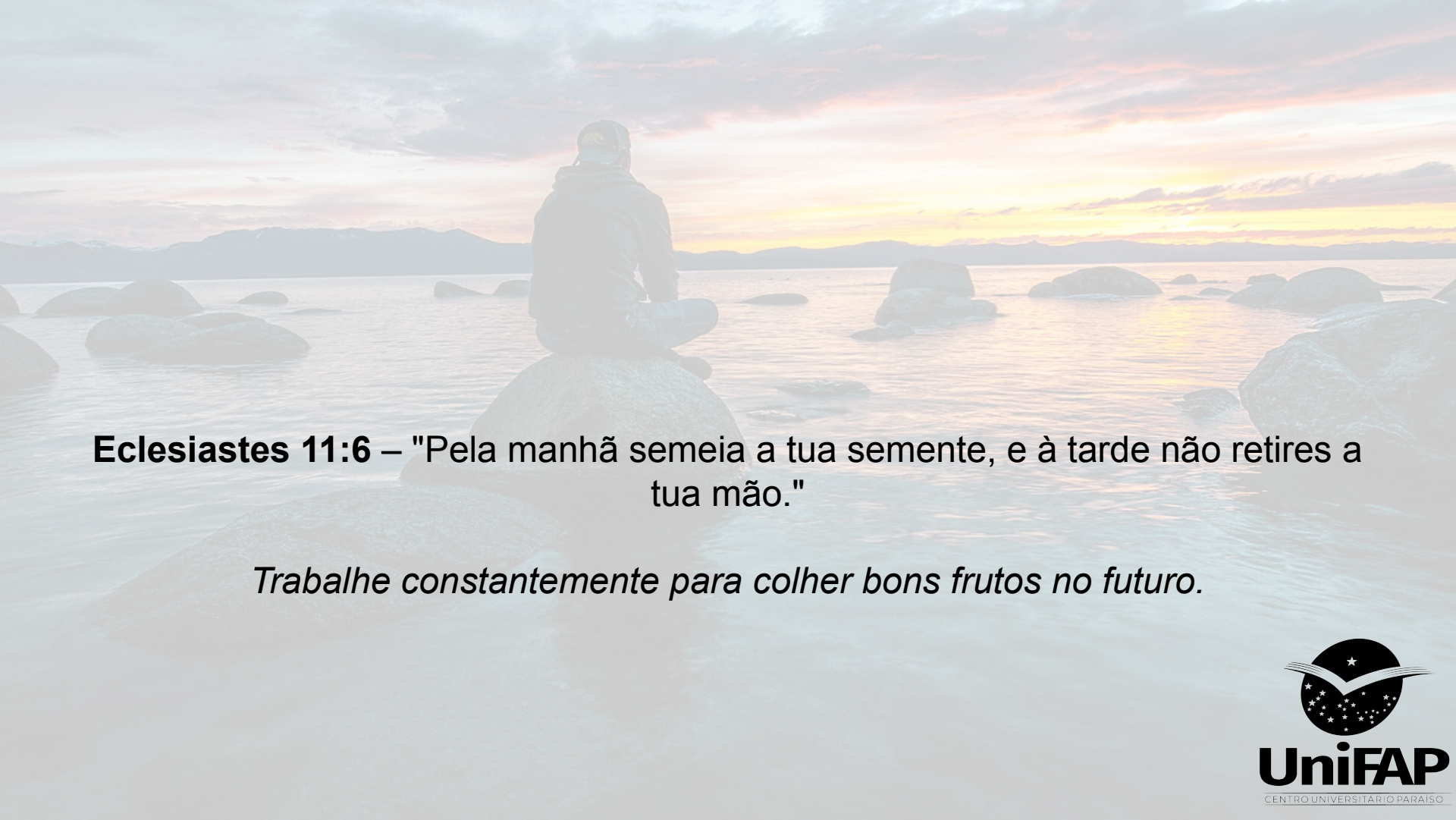




Perguntas?!



UniFAP
CENTRO UNIVERSITÁRIO PARAÍSO



Eclesiastes 11:6 – "Pela manhã semeia a tua semente, e à tarde não retires a tua mão."

Trabalhe constantemente para colher bons frutos no futuro.

UniFAP
CENTRO UNIVERSITÁRIO PARAÍSO



Obrigado!

UniFAP
CENTRO UNIVERSITÁRIO PARAÍSO