

Vamos detalhar o **Passo 2**, que foca na definição do design de interface usando **XML** para criar a estrutura visual do aplicativo. Aqui, vamos criar as telas principais que envolvem o fluxo básico do aplicativo de barbearia.

2. Definição do Design de Interface (XML)

2.1 Tela de Login

A tela de login permite que os usuários (clientes ou barbeiros) façam login no sistema. Vamos usar componentes como `EditText` para entrada de dados e `Button` para ações de login.

Código XML da Tela de Login (`activity_login.xml`):

xml

 Copiar código

```
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools" android:layout_width="match_parent"
android:layout_height="match_parent" tools:context=".LoginActivity"> <TextView
android:id="@+id/titleTextView" android:layout_width="wrap_content"
android:layout_height="wrap_content" android:text="Login" android:textSize="24sp"
android:layout_marginTop="50dp" app:layout_constraintTop_toTopOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintEnd_toEndOf="parent"/> <EditText android:id="@+id/emailEditText"
android:layout_width="0dp" android:layout_height="wrap_content" android:hint="Email"
android:inputType="textEmailAddress" android:layout_marginTop="20dp"
app:layout_constraintTop_toBottomOf="@id/titleTextView"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintEnd_toEndOf="parent" android:layout_marginHorizontal="32dp"/>
<EditText android:id="@+id/passwordEditText" android:layout_width="0dp"
android:layout_height="wrap_content" android:hint="Senha"
android:inputType="textPassword" android:layout_marginTop="20dp"
app:layout_constraintTop_toBottomOf="@id/emailEditText"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintEnd_toEndOf="parent" android:layout_marginHorizontal="32dp"/>
<Button android:id="@+id/loginButton" android:layout_width="wrap_content"
android:layout_height="wrap_content" android:text="Login"
android:layout_marginTop="20dp"
app:layout_constraintTop_toBottomOf="@id/passwordEditText"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintEnd_toEndOf="parent"/> <TextView
android:id="@+id/registerTextView" android:layout_width="wrap_content"
android:layout_height="wrap_content" android:text="Não tem uma conta? Cadastre-se"
android:textSize="14sp" android:textColor="@color/purple_500"
android:layout_marginTop="20dp" app:layout_constraintTop_toBottomOf="@id/loginButton"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintEnd_toEndOf="parent"/>
</androidx.constraintlayout.widget.ConstraintLayout>
```

Explicação:

- EditText : Usado para inserir email e senha.
- Button : Para enviar as credenciais e efetuar login.
- TextView : Para linkar a navegação ao cadastro de novo usuário.

2.2 Tela de Cadastro de Cliente/Barbeiro

Vamos criar uma tela de cadastro para que novos clientes e barbeiros possam se registrar.

Código XML da Tela de Cadastro (activity_register.xml):

xml

 Copiar código

```
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools" android:layout_width="match_parent"
android:layout_height="match_parent" tools:context=".RegisterActivity"> <TextView
android:id="@+id/titleTextView" android:layout_width="wrap_content"
android:layout_height="wrap_content" android:text="Cadastro" android:textSize="24sp"
android:layout_marginTop="50dp" app:layout_constraintTop_toTopOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintEnd_toEndOf="parent"/> <EditText android:id="@+id/nameEditText"
android:layout_width="0dp" android:layout_height="wrap_content" android:hint="Nome
Completo" android:layout_marginTop="20dp"
app:layout_constraintTop_toBottomOf="@id/titleTextView"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintEnd_toEndOf="parent" android:layout_marginHorizontal="32dp"/>
<EditText android:id="@+id/emailEditText" android:layout_width="0dp"
android:layout_height="wrap_content" android:hint="Email"
android:inputType="textEmailAddress" android:layout_marginTop="20dp"
app:layout_constraintTop_toBottomOf="@id/nameEditText"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintEnd_toEndOf="parent" android:layout_marginHorizontal="32dp"/>
<EditText android:id="@+id/passwordEditText" android:layout_width="0dp"
android:layout_height="wrap_content" android:hint="Senha"
android:inputType="textPassword" android:layout_marginTop="20dp"
app:layout_constraintTop_toBottomOf="@id/emailEditText"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintEnd_toEndOf="parent" android:layout_marginHorizontal="32dp"/>
<Button android:id="@+id/registerButton" android:layout_width="wrap_content"
android:layout_height="wrap_content" android:text="Cadastrar"
android:layout_marginTop="20dp"
app:layout_constraintTop_toBottomOf="@id/passwordEditText"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintEnd_toEndOf="parent"/>
</androidx.constraintlayout.widget.ConstraintLayout>
```

Explicação:

- EditText : Campos para nome, email e senha.
- Button : Para submeter o formulário de cadastro.

2.3 Tela de Agendamento de Serviço

Esta tela permite que o cliente escolha o serviço, data e horário. Vamos utilizar `Spinner` para selecionar o serviço e um `DatePicker` para selecionar a data.

Código XML da Tela de Agendamento (`activity_schedule.xml`):

xml

 Copiar código

```
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools" android:layout_width="match_parent"
android:layout_height="match_parent" tools:context=".ScheduleActivity"> <TextView
android:id="@+id/titleTextView" android:layout_width="wrap_content"
android:layout_height="wrap_content" android:text="Agendar Serviço"
android:textSize="24sp" android:layout_marginTop="50dp"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintEnd_toEndOf="parent"/> <Spinner android:id="@+id/serviceSpinner"
android:layout_width="0dp" android:layout_height="wrap_content"
android:layout_marginTop="20dp"
app:layout_constraintTop_toBottomOf="@id/titleTextView"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintEnd_toEndOf="parent" android:layout_marginHorizontal="32dp" />
<DatePicker android:id="@+id/datePicker" android:layout_width="wrap_content"
android:layout_height="wrap_content" android:layout_marginTop="20dp"
app:layout_constraintTop_toBottomOf="@id/serviceSpinner"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintEnd_toEndOf="parent"/> <Button android:id="@+id/scheduleButton"
android:layout_width="wrap_content" android:layout_height="wrap_content"
android:text="Agendar" android:layout_marginTop="20dp"
app:layout_constraintTop_toBottomOf="@id/datePicker"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintEnd_toEndOf="parent"/>
</androidx.constraintlayout.widget.ConstraintLayout>
```

Explicação:

- Spinner : Permite a escolha do serviço (corte, barba, etc.).
- DatePicker : Componente de data para selecionar o dia do serviço.
- Button : Para confirmar o agendamento.

2.4 Tela de Histórico de Agendamentos

Esta tela mostra uma lista de serviços agendados. Vamos utilizar o `RecyclerView` para exibir uma lista dinâmica de itens.

Código XML da Tela de Histórico (activity_history.xml):

xml

 Copiar código

```
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools" android:layout_width="match_parent"
android:layout_height="match_parent" tools:context=".HistoryActivity"> <TextView
android:id="@+id/titleTextView" android:layout_width="wrap_content"
android:layout_height="wrap_content" android:text="Histórico de Agendamentos"
android:textSize="24sp" android:layout_marginTop="50dp"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintEnd_toEndOf="parent"/> <androidx.recyclerview.widget.RecyclerView
android:id="@+id/historyRecyclerView" android:layout_width="0dp"
android:layout_height="0dp" android:layout_marginTop="20dp"
app:layout_constraintTop_toBottomOf="@id/titleTextView"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintBottom_toBottomOf="parent"
android:layout_marginHorizontal="16dp"/>
</androidx.constraintlayout.widget.ConstraintLayout>
```

Explicação:

- RecyclerView : Exibe uma lista de agendamentos que foram feitos pelo cliente.

2.5 Conclusão do Design

Com essas telas básicas (Login, Cadastro, Agendamento, Histórico), você terá a estrutura visual para o aplicativo de barbearia. A partir daqui, é possível integrar a lógica de backend com o Kotlin e armazenar dados utilizando ferramentas como Room ou Firebase.

Esses layouts são apenas a base; você pode personalizar o design e adicionar mais funcionalidades à medida que o desenvolvimento avança.