

Anahí Salgado

@anncode





Android Jetpack

Jetpack

[OVERVIEW](#)
[GET STARTED](#)
[LIBRARIES](#)
[COMMUNITY](#)
[COMPOSE](#)

Join us for [#Android11: The Beta Launch Show](#) on June 3!

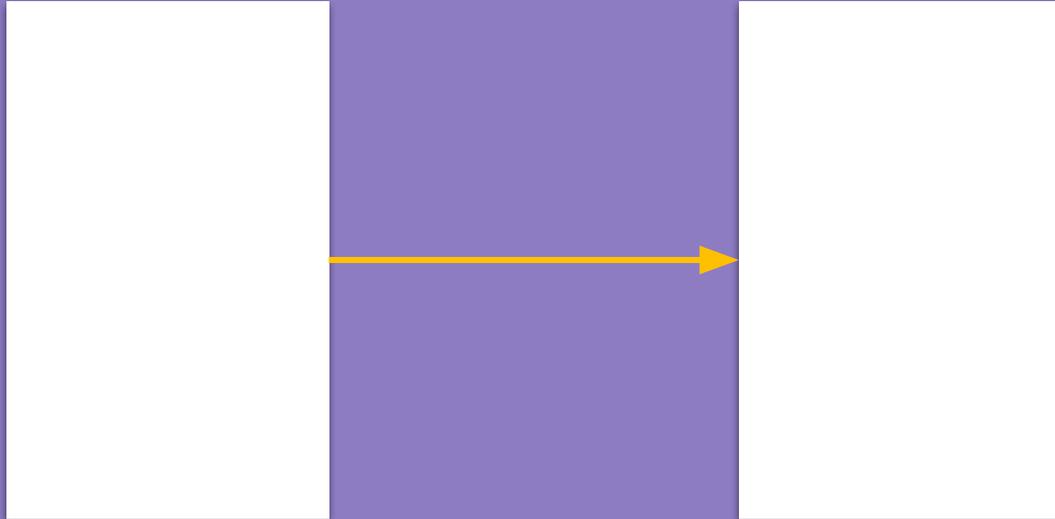
Android Jetpack

Jetpack is a suite of libraries, tools, and guidance to help developers write high-quality apps more easily. These components help you follow best practices, free you from writing boilerplate code, and simplify complex tasks, so you can focus on the code you care about.

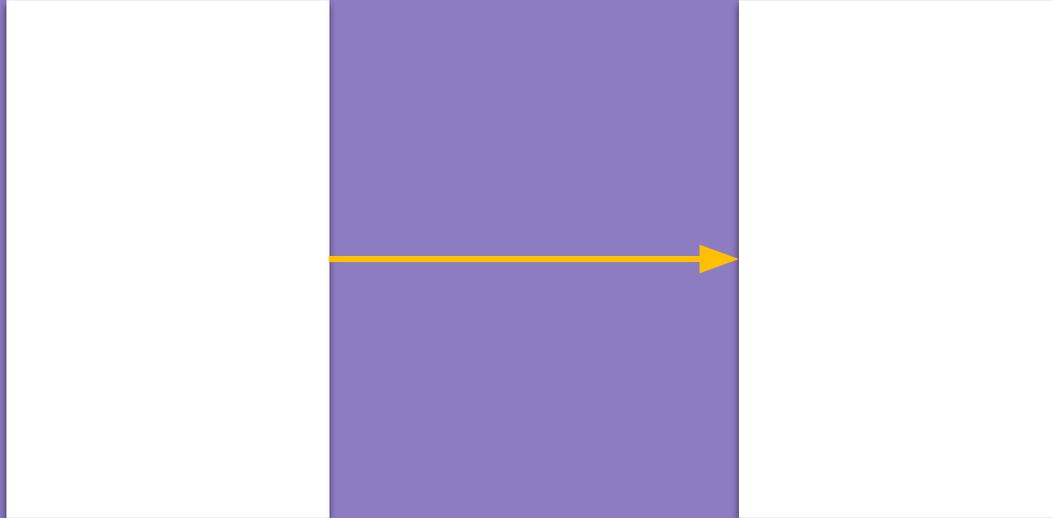
Jetpack comprises the [androidx.*](#) package libraries, unbundled from the platform APIs. This means that it offers backward compatibility and is updated more frequently than the Android platform, making sure you always have access to the latest and greatest versions of the Jetpack components.



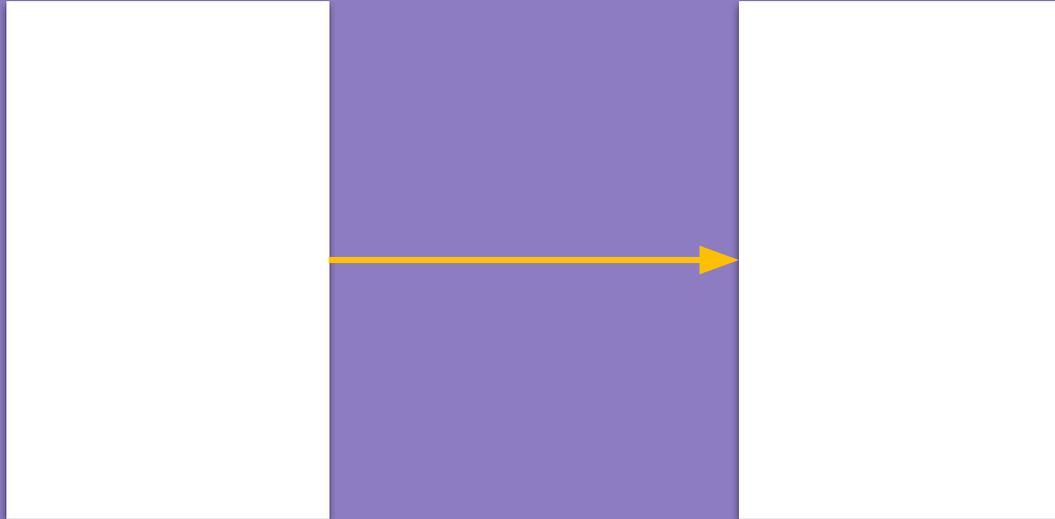
[@anncode](#)



```
val intent = Intent(this, MyActivity::class.java)
intent.putExtra("name", user.name)
startActivity(intent)
```



```
val intent = Intent(this, MyActivity::class.java)
intent.putExtra("name", user.name)
startActivity(intent)
```



```
val fragment = MyFragment()
fragment.arguments = Bundle().apply { putString("name", user.name) }
requireFragmentManager().beginTransaction()
    .addToBackStack(null)
    .replace(R.id.main_content, fragment)
    .commit()
```

```
val intent = Intent(this, MyActivity::class.java)
intent.putExtra("name", user.name)
startActivity(intent)
```



```
val fragment = MyFragment()
fragment.arguments = Bundle().apply { putString("name", user.name) }
requireFragmentManager().beginTransaction()
    .addToBackStack(null)
    .replace(R.id.main_content, fragment)
    .commit()
```

```
val intent = Intent(this, MyActivity::class.java)
intent.putExtra("name", user.name)
startActivity(intent)
```

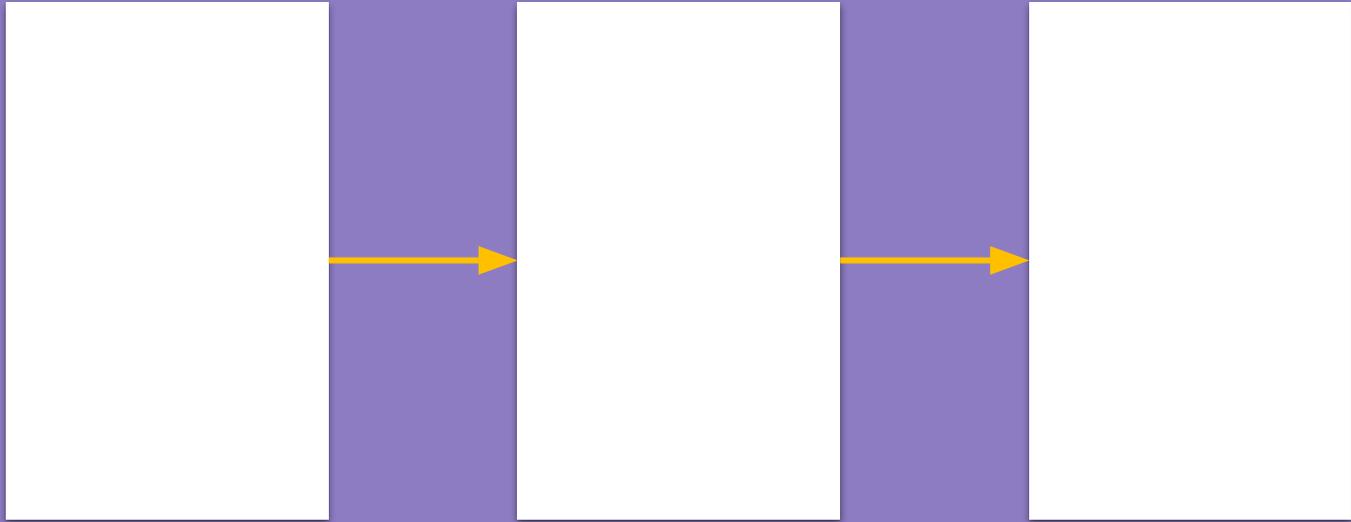
```
val fragment = MyFragment()
fragment.arguments = Bundle().apply { putString("name", user.name) }
requireFragmentManager().beginTransaction()
    .addToBackStack(null)
    .replace(R.id.main_content, fragment)
    .commit()
```

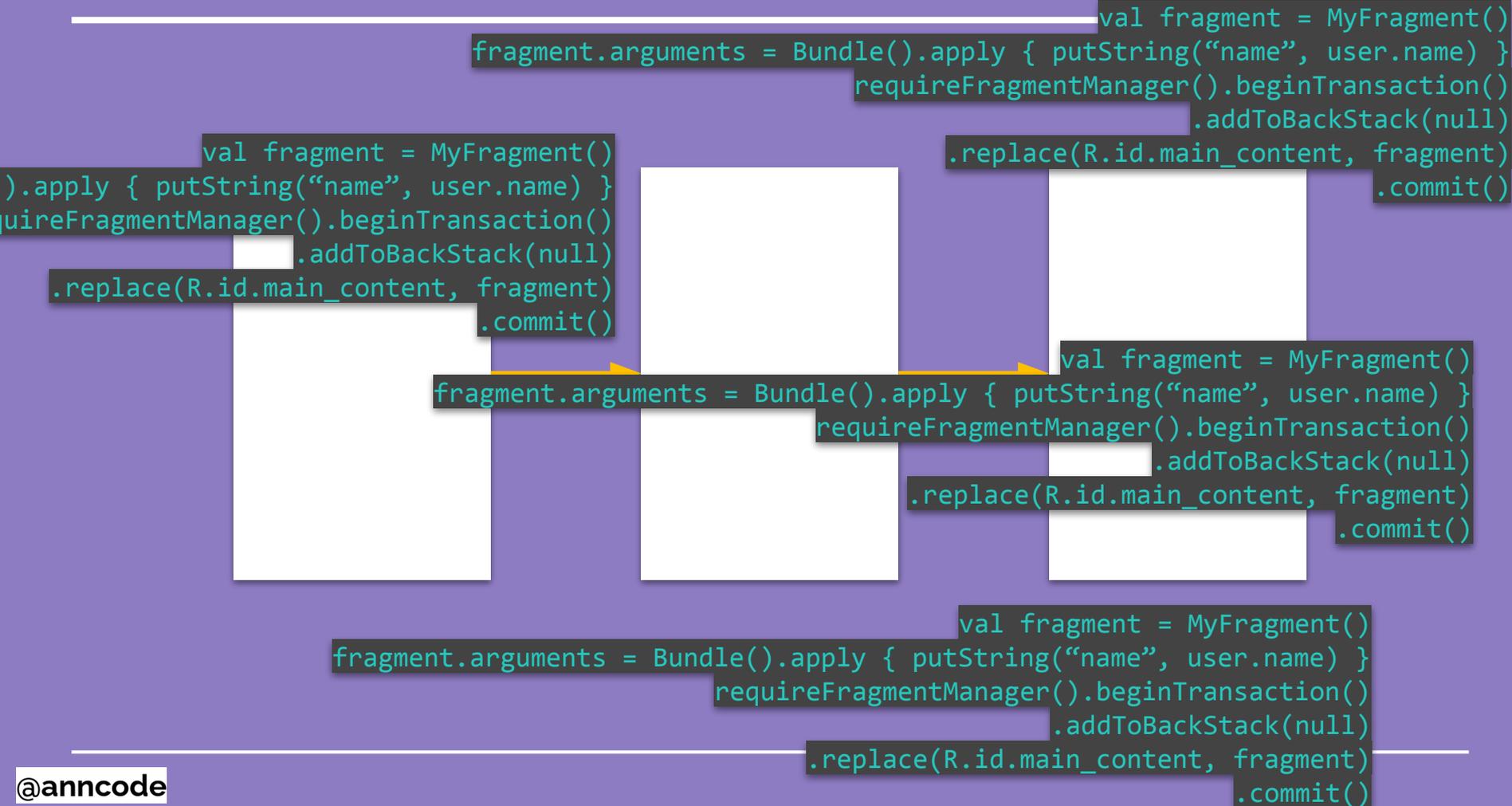
```
        val fragment = MyFragment()
        fragment.arguments = Bundle().apply { putString("name", user.name) }
        requireFragmentManager().beginTransaction()
            .addToBackStack(null)
            .replace(R.id.main_content, fragment)
            .commit()
```

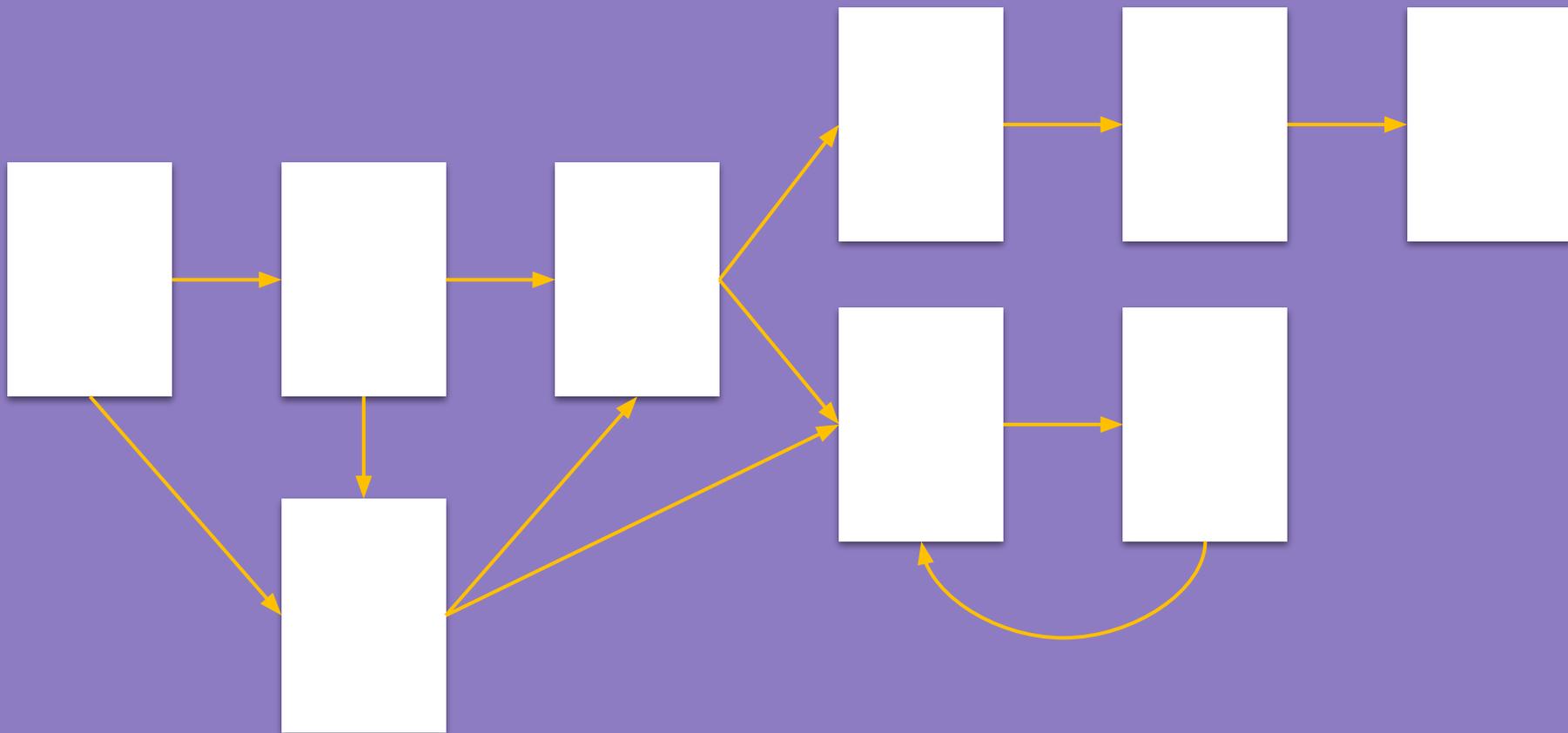
```
        val fragment = MyFragment()
        ).apply { putString("name", user.name) }
        requireFragmentManager().beginTransaction()
            .addToBackStack(null)
            .replace(R.id.main_content, fragment)
            .commit()
```

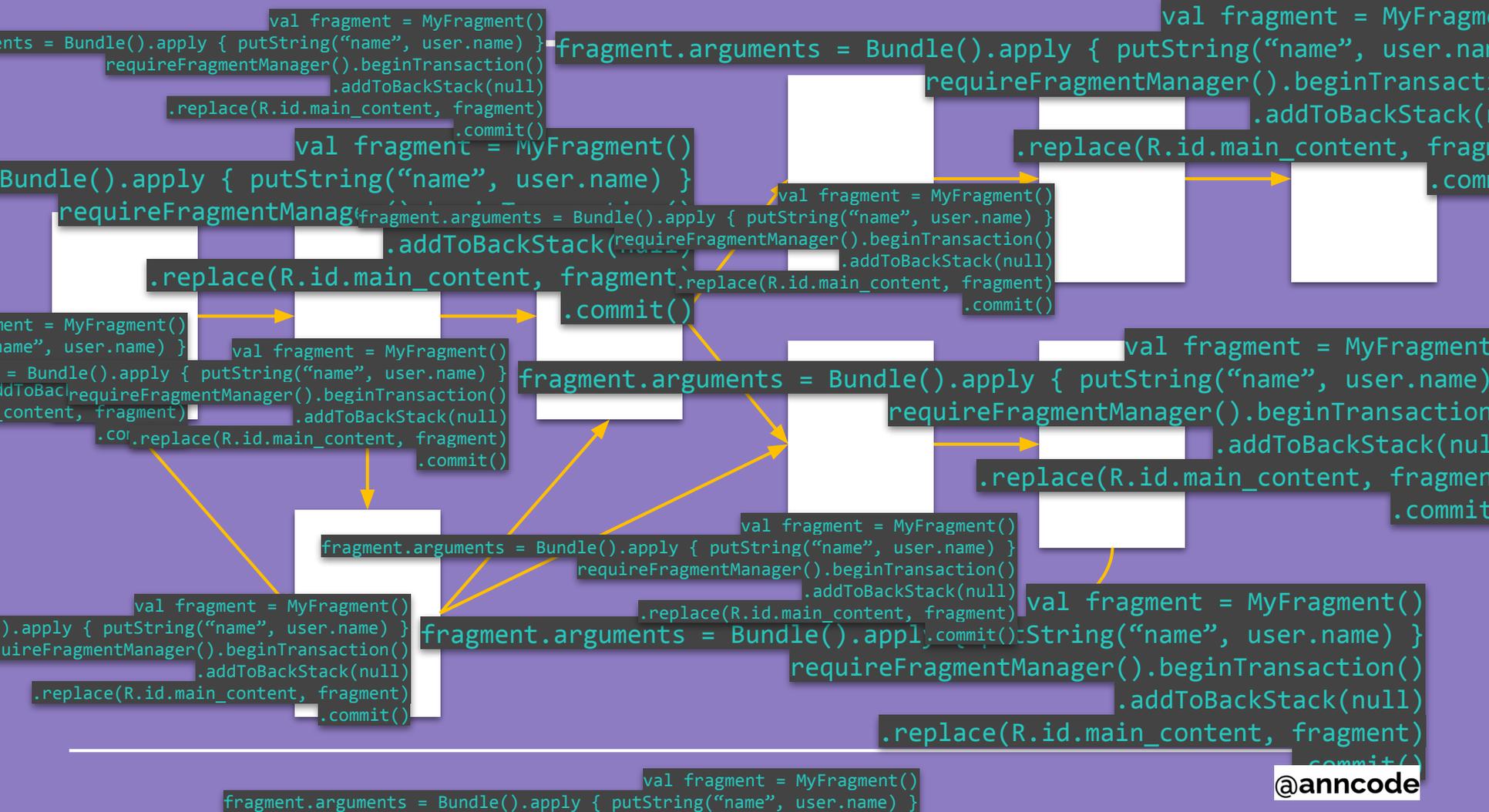
```
        val fragment = MyFragment()
        fragment.arguments = Bundle().apply { putString("name", user.name) }
        requireFragmentManager().beginTransaction()
            .addToBackStack(null)
            .replace(R.id.main_content, fragment)
            .commit()
```

```
        val fragment = MyFragment()
        fragment.arguments = Bundle().apply { putString("name", user.name) }
        requireFragmentManager().beginTransaction()
            .addToBackStack(null)
            .replace(R.id.main_content, fragment)
            .commit()
```



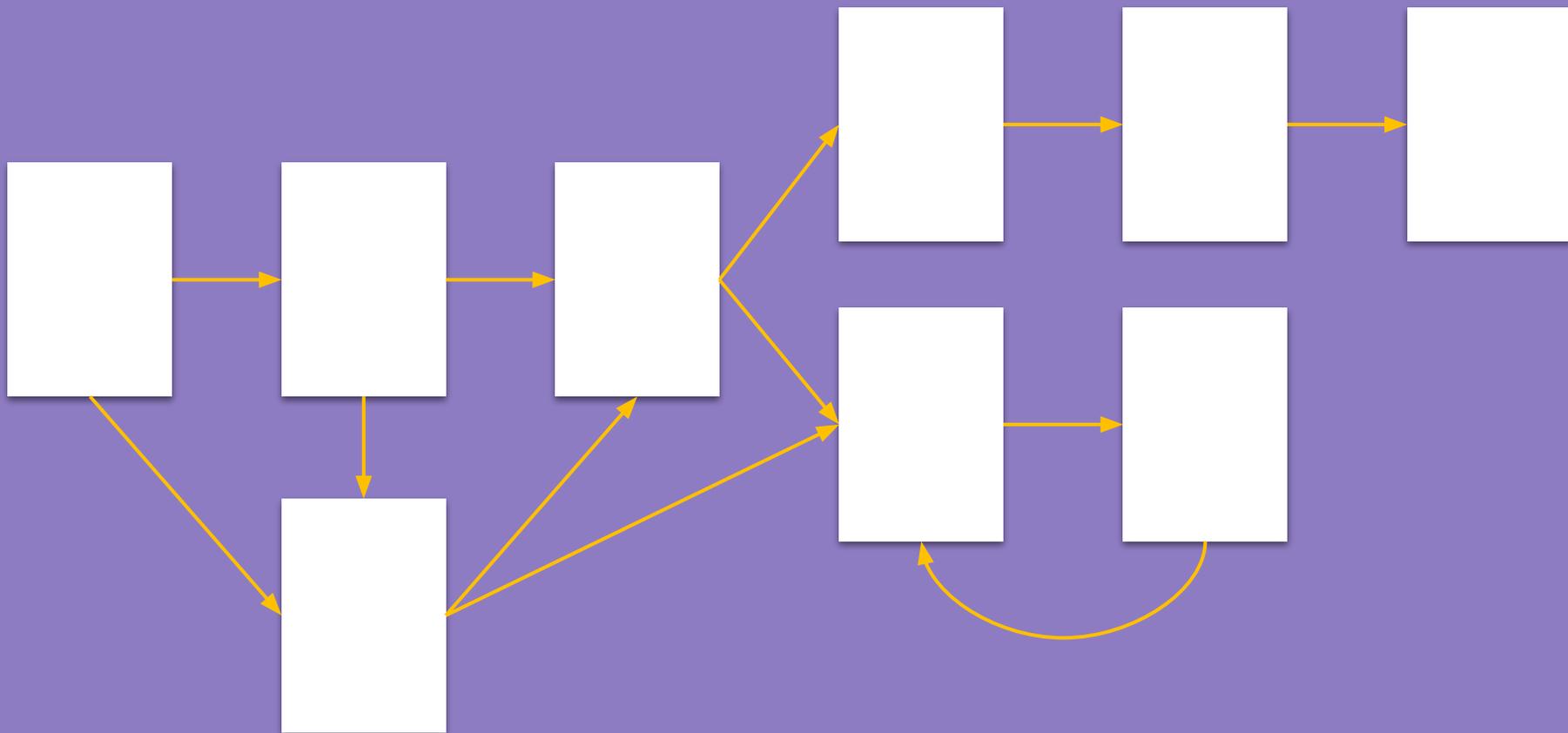


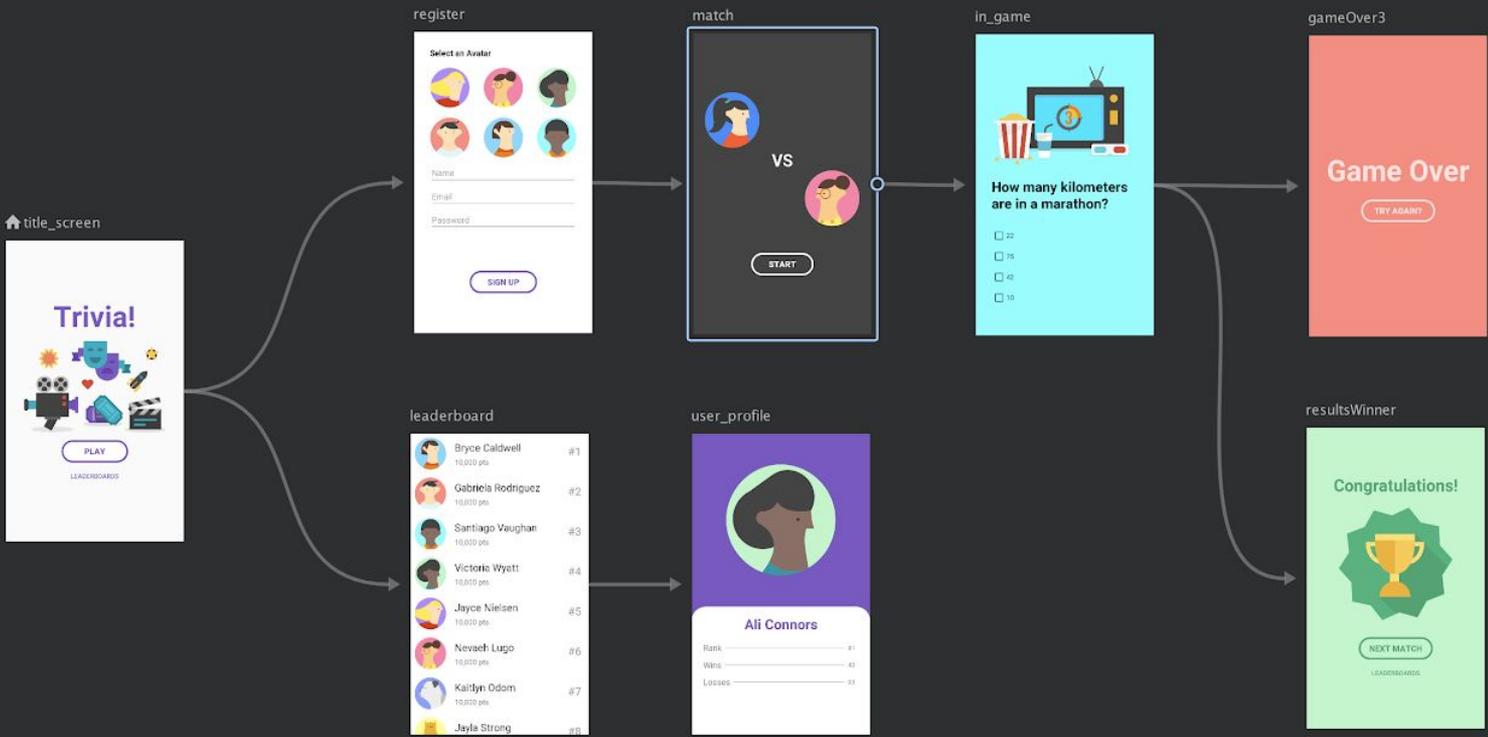




Android Navigation Component







Attributes

Type:

Label:

ID:

Class:

Arguments +
Click + to add Arguments

Actions +
→ in_game (action_match_to_in_game)

Deep Links +
Click + to add Deep Links

Android Navigation Component

Navegación escalable y clara ✓

Android Navigation Component

Navegación escalable y clara ✓

Boilerplate code ✓

Android Navigation Component

Navegación escalable y clara ✓

Boilerplate code ✓

Clean Code and Architecture ✓

Android Navigation Component

```
dependencies {  
    def nav_version = "2.2.2"  
  
    // Java language implementation  
    implementation "androidx.navigation:navigation-fragment:$nav_version"  
    implementation "androidx.navigation:navigation-ui:$nav_version"  
  
    // Kotlin  
    implementation "androidx.navigation:navigation-fragment-ktx:$nav_version"  
    implementation "androidx.navigation:navigation-ui-ktx:$nav_version"  
}
```

Safe Args

Project build.gradle

```
dependencies {  
    classpath "androidx.navigation:navigation-safe-args-gradle-plugin:$nav_version"  
}
```

Module build.gradle

```
apply plugin: 'androidx.navigation.safeargs.kotlin'
```

Android

app

- manifests
- java
 - com
 - anncode
 - androidnavigationcomponent
 - MainActivity
- com (androidTest)
- com (test)

```
30 implementation "org.jetbrains.kotlin:kotlin-stdlib-jdk7:$kotlin_ver
31 implementation 'androidx.appcompat:appcompat:1.1.0'
32 implementation 'androidx.core:core-ktx:1.2.0'
33 implementation 'androidx.constraintlayout:constraintlayout:1.1.3'
34 testImplementation 'junit:junit:4.12'
35 androidTestImplementation 'androidx.test.ext:junit:1.1.1'
```

New Resource File

File name:

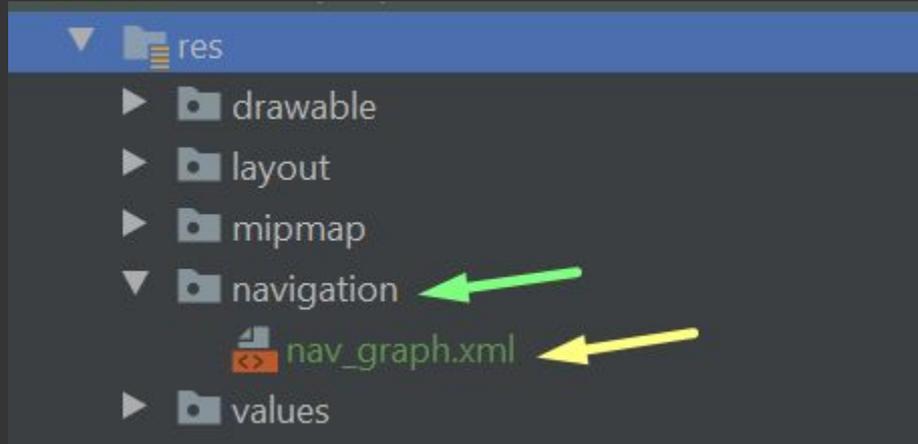
Resource type:

Root element:

Source set:

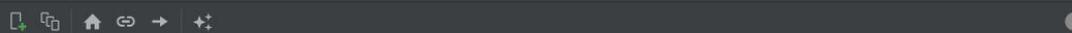
Directory name:

OK Cancel



Destinations

HOST
activity_main (nav_host_fragment)
GRAPH



Click to add a destination

Attributes

nav_graph		navigation
id	nav_graph	
label		
startDestination	none	
▶ Argument Default Values		
▶ Global Actions + -		
▼ Deep Links + -		

Nothing to show

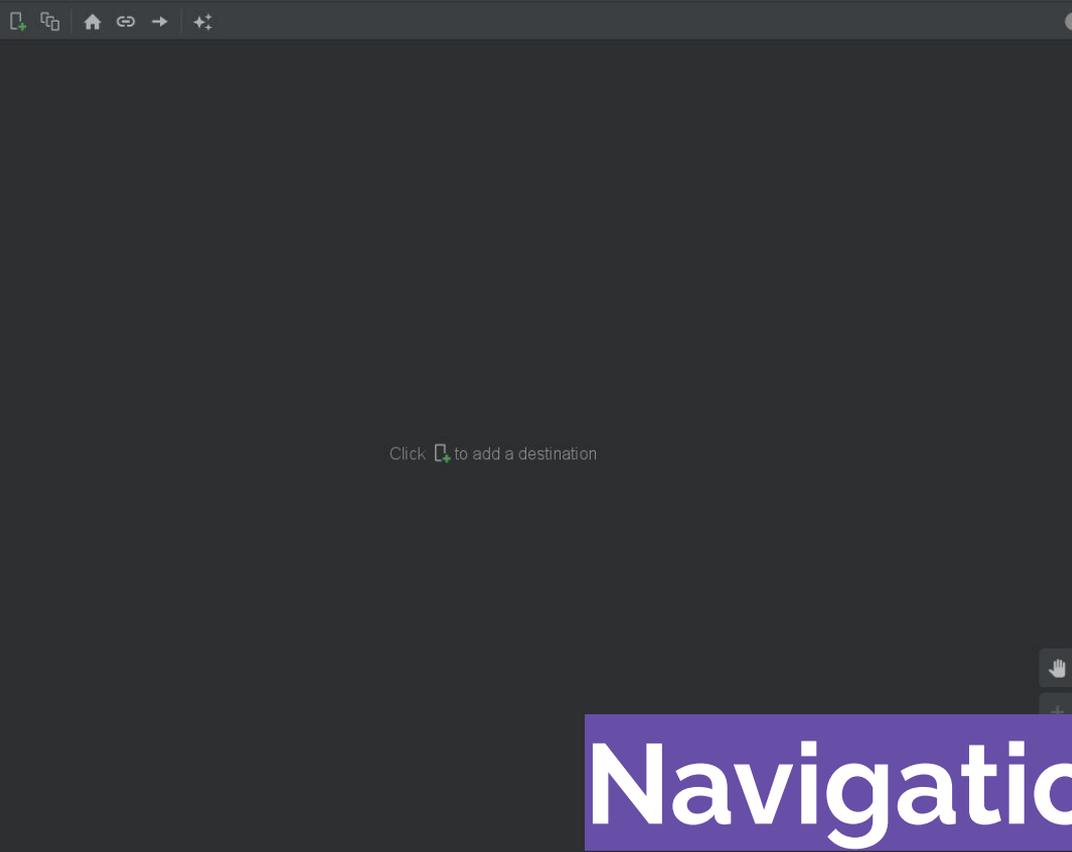


Destinations

HOST

activity_main (nav_host_fragment)

GRAPH



Attributes

nav_graph navigation

id nav_graph

label

startDestination none

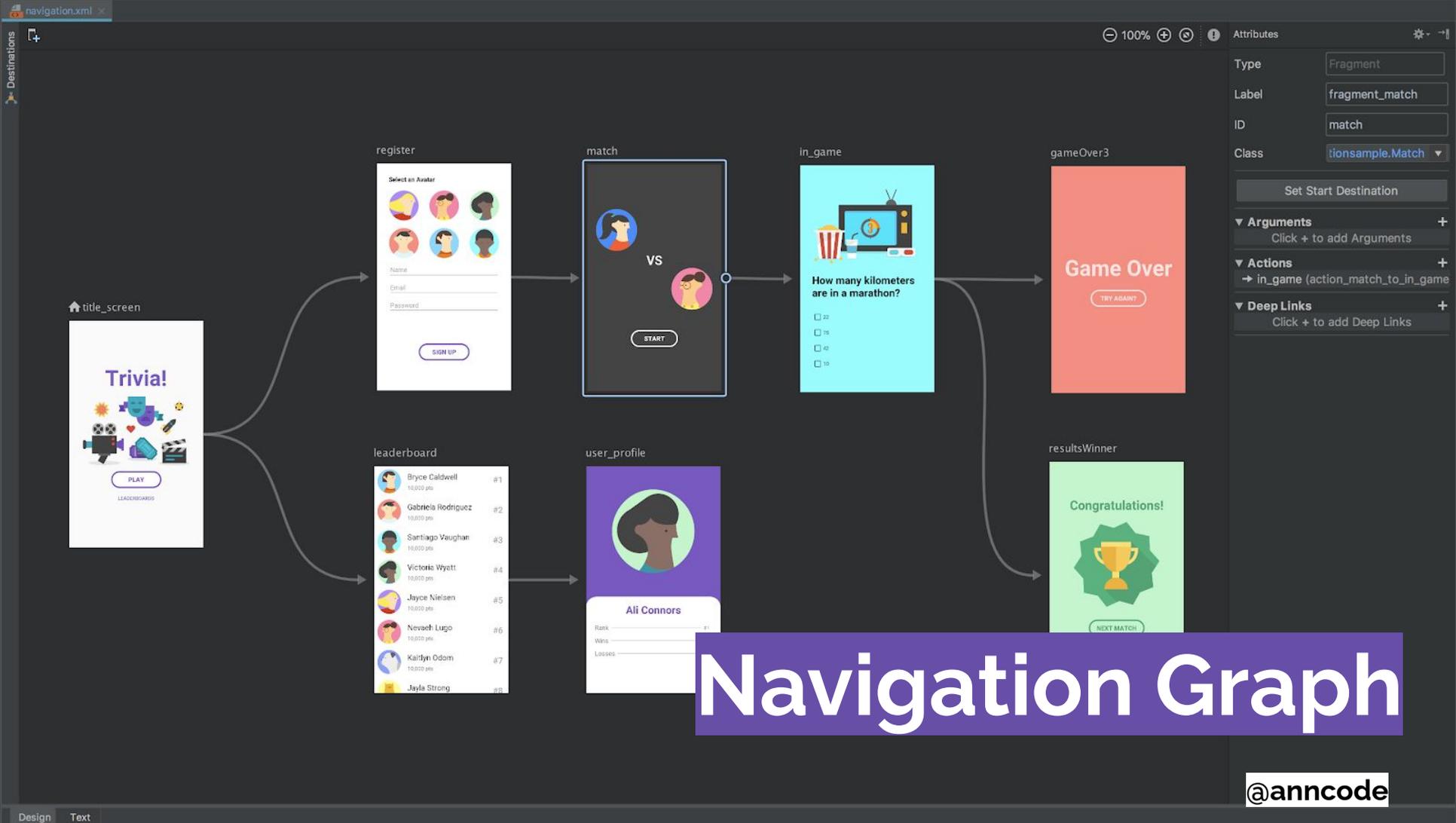
Argument Default Values

Global Actions + -

Deep Links + -

Nothing to show

Navigation Graph

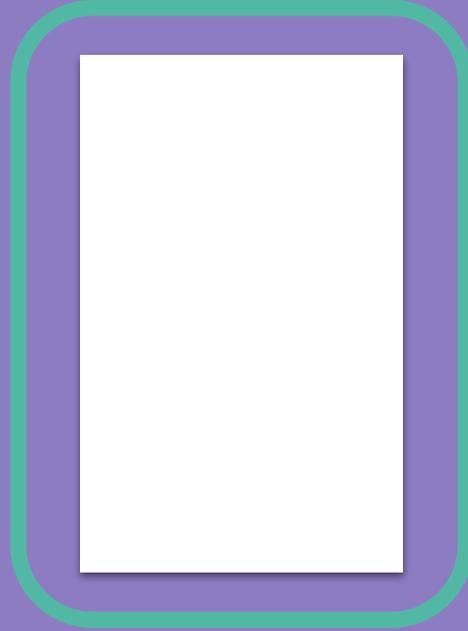


Navigation Graph

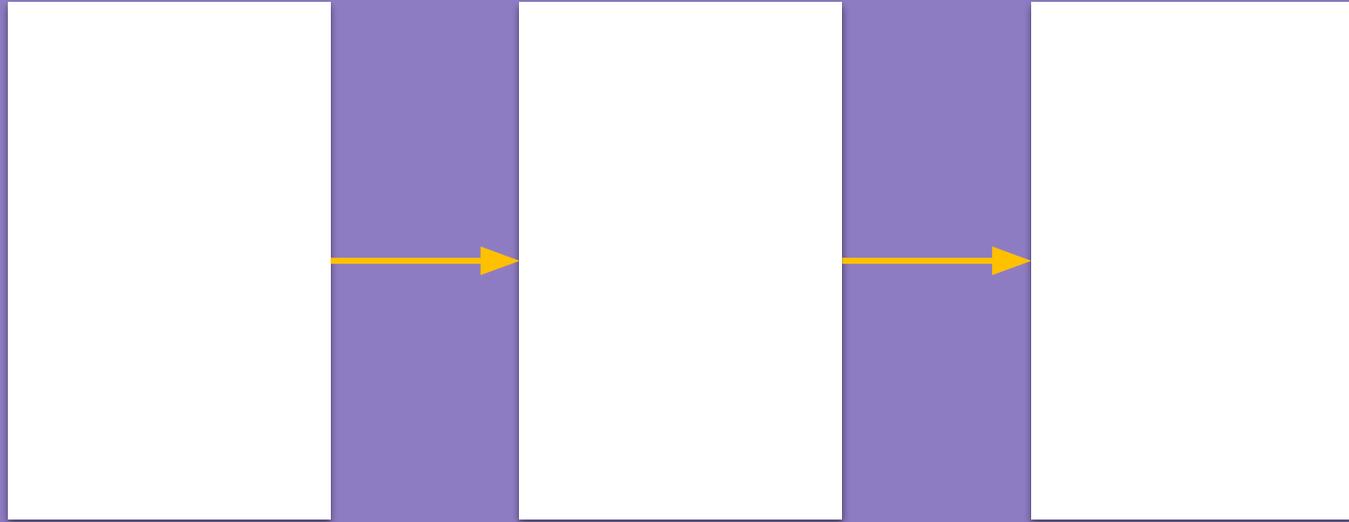
Destinations



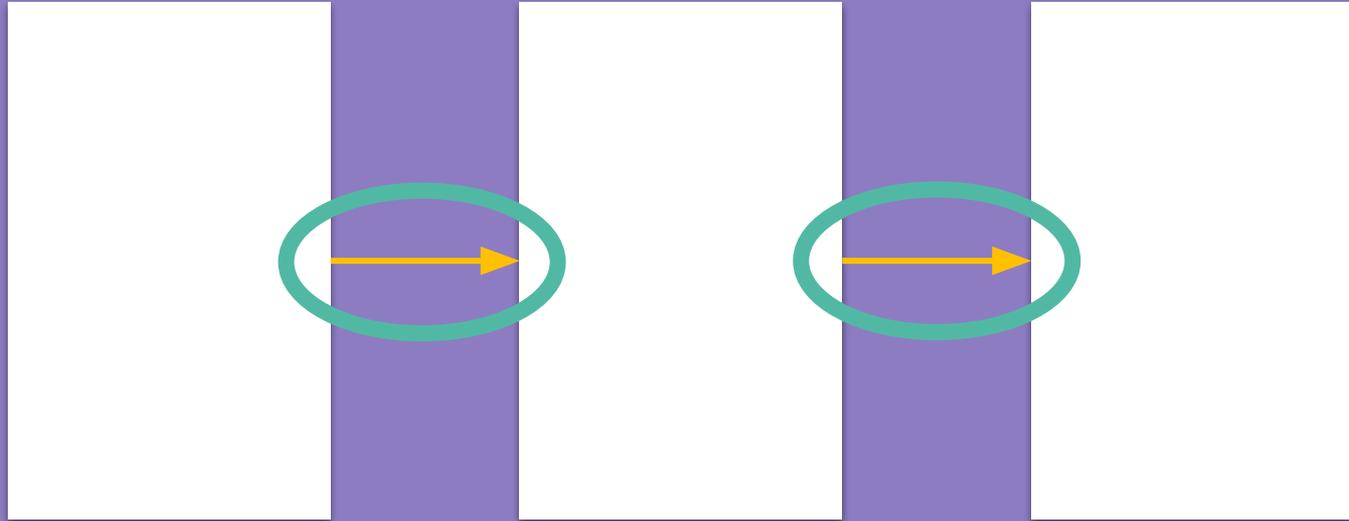
Destinations

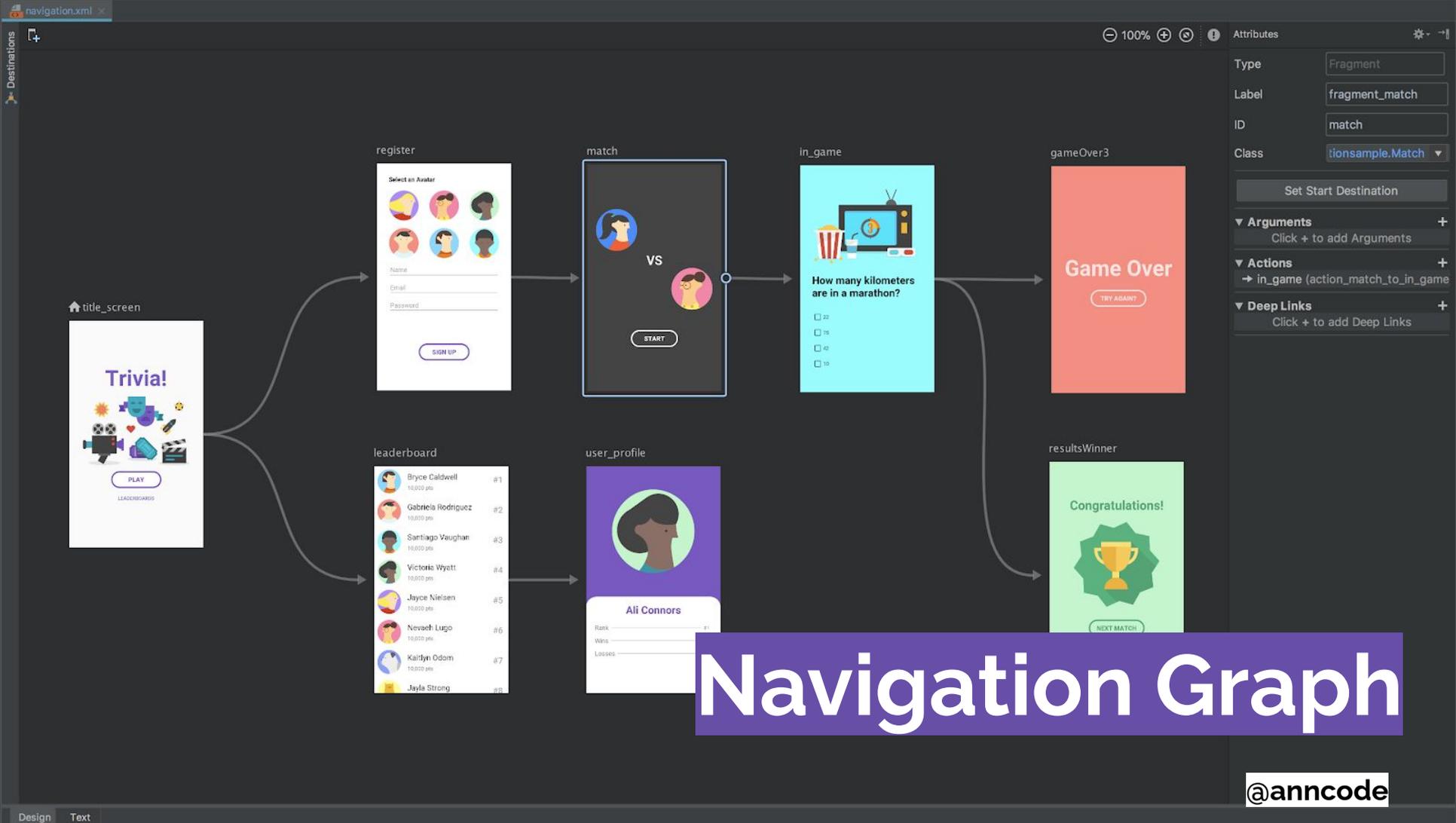


Actions



Actions





Type:
Label:
ID:
Class: [Dropdown]
[Set Start Destination]
▼ Arguments [+]
Click + to add Arguments
▼ Actions [+]
→ in_game (action_match_to_in_game)
▼ Deep Links [+]
Click + to add Deep Links

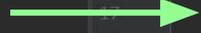
Navigation Graph

Destination

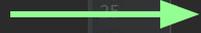


```
1 <?xml version="1.0" encoding="utf-8"?>
2 <navigation xmlns:android="http://schemas.android.com/apk/res/android"
3   xmlns:app="http://schemas.android.com/apk/res-auto"
4   xmlns:tools="http://schemas.android.com/tools"
5   android:id="@+id/nav_graph"
6   app:startDestination="@id/loginFragment">
7   <fragment
8     android:id="@+id/loginFragment"
9     android:name="com.anncode.androidnavigationcomponent.LoginFragment"
10    android:label="LoginFragment"
11    tools:layout="@layout/fragment_login">
12
13    <action android:id="@+id/toCreateAccountFragment" app:destination="@id/createAccountFragment" />
14
15    <action android:id="@+id/toHomeFragment" app:destination="@id/homeFragment" />
16  </fragment>
17  <fragment
18    android:id="@+id/createAccountFragment"
19    android:name="com.anncode.androidnavigationcomponent.CreateAccountFragment"
20    android:label="CreateAccountFragment"
21    tools:layout="@layout/fragment_create_account">
22
23    <action android:id="@+id/toLoginFragment" app:destination="@id/loginFragment" />
24  </fragment>
25  <fragment
26    android:id="@+id/homeFragment"
27    android:name="com.anncode.androidnavigationcomponent.HomeFragment"
28    android:label="HomeFragment"
29    tools:layout="@layout/fragment_home"/>
30 </navigation>
```

Destination



Destination



Action

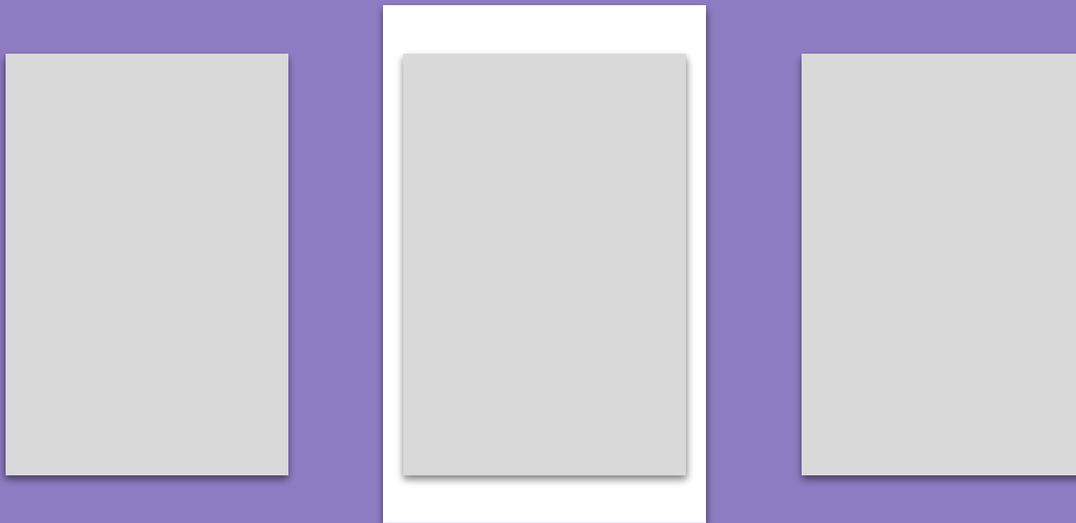
Action

```
nav_graph.xml
1 <?xml version="1.0" encoding="utf-8"?>
2 <navigation xmlns:android="http://schemas.android.com/apk/res/android"
3   xmlns:app="http://schemas.android.com/apk/res-auto"
4   xmlns:tools="http://schemas.android.com/tools"
5   android:id="@+id/nav_graph"
6   app:startDestination="@id/loginFragment">
7   <fragment
8     android:id="@+id/loginFragment"
9     android:name="com.anncode.androidnavigationcomponent.LoginFragment"
10    android:label="LoginFragment"
11    tools:layout="@layout/fragment_login">
12
13    <action android:id="@+id/toCreateAccountFragment" app:destination="@id/createAccountFragment" />
14
15    <action android:id="@+id/toHomeFragment" app:destination="@id/homeFragment" />
16  </fragment>
17  <fragment
18    android:id="@+id/createAccountFragment"
19    android:name="com.anncode.androidnavigationcomponent.CreateAccountFragment"
20    android:label="CreateAccountFragment"
21    tools:layout="@layout/fragment_create_account">
22
23    <action android:id="@+id/toLoginFragment" app:destination="@id/loginFragment" />
24  </fragment>
25  <fragment
26    android:id="@+id/homeFragment"
27    android:name="com.anncode.androidnavigationcomponent.HomeFragment"
28    android:label="HomeFragment"
29    tools:layout="@layout/fragment_home"/>
30 </navigation>
```

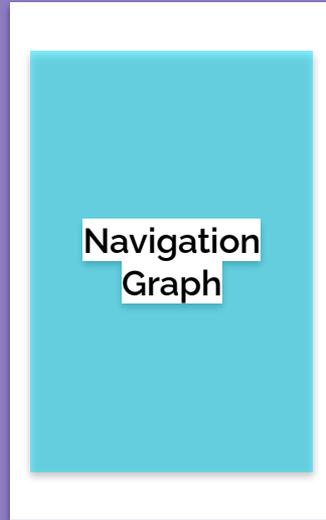
MainActivity



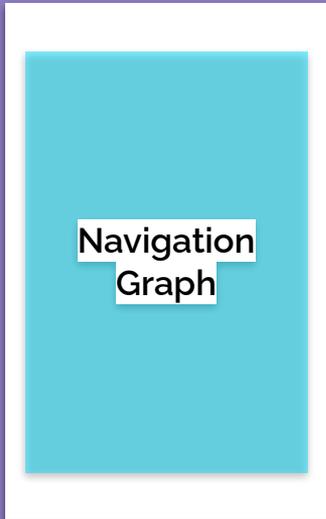
MainActivity



MainActivity

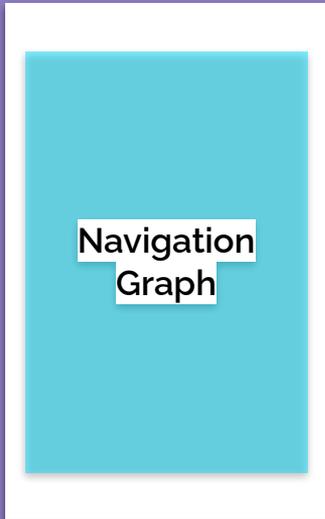


MainActivity



```
<fragment
    android:id="@+id/nav_host_fragment"
    android:name="androidx.navigation.fragment.NavHostFragment"
    android:layout_width="0dp"
    android:layout_height="0dp"
    app:defaultNavHost="true"
    app:navGraph="@navigation/nav_graph" />
```

MainActivity



```
<fragment
    android:id="@+id/nav_host_fragment"
    android:name="androidx.navigation.fragment.NavHostFragment"
    android:layout_width="0dp"
    android:layout_height="0dp"
    app:defaultNavHost="true"
    app:navGraph="@navigation/nav_graph" />
```

NavHostFragment

XML

Esquema del flujo de navegación

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <navigation xmlns:android="http://schemas.android.com/apk/res/android"
3     xmlns:app="http://schemas.android.com/apk/res-auto"
4     xmlns:tools="http://schemas.android.com/tools"
5     android:id="@+id/nav_graph"
6     app:startDestination="@id/loginFragment">
7     <fragment
8         android:id="@+id/loginFragment"
9         android:name="com.anncode.androidnavigationcomponent.LoginFragment"
10        android:label="LoginFragment"
11        tools:layout="@layout/fragment_login">
12
13        <action android:id="@+id/toCreateAccountFragment" app:destination="@id/createAccountFragment" />
14
15        <action android:id="@+id/toHomeFragment" app:destination="@id/homeFragment" />
16    </fragment>
17    <fragment
18        android:id="@+id/createAccountFragment"
19        android:name="com.anncode.androidnavigationcomponent.CreateAccountFragment"
20        android:label="CreateAccountFragment"
21        tools:layout="@layout/fragment_create_account">
22
23        <action android:id="@+id/toLoginFragment" app:destination="@id/loginFragment" />
24    </fragment>
25    <fragment
26        android:id="@+id/homeFragment"
27        android:name="com.anncode.androidnavigationcomponent.HomeFragment"
28        android:label="HomeFragment"
29        tools:layout="@layout/fragment_home"/>
30 </navigation>
```

XML

Esquema del flujo de navegación

XML

**Esquema del flujo de
navegación**

Kotlin

**OnClickListener
llama a los actions**

Clases
Autogeneradas

XML

**Esquema del flujo de
navegación**

Kotlin

**OnClickListener
llama a los actions**

NavHostFragment

NavHostFragment



NavHostFragment

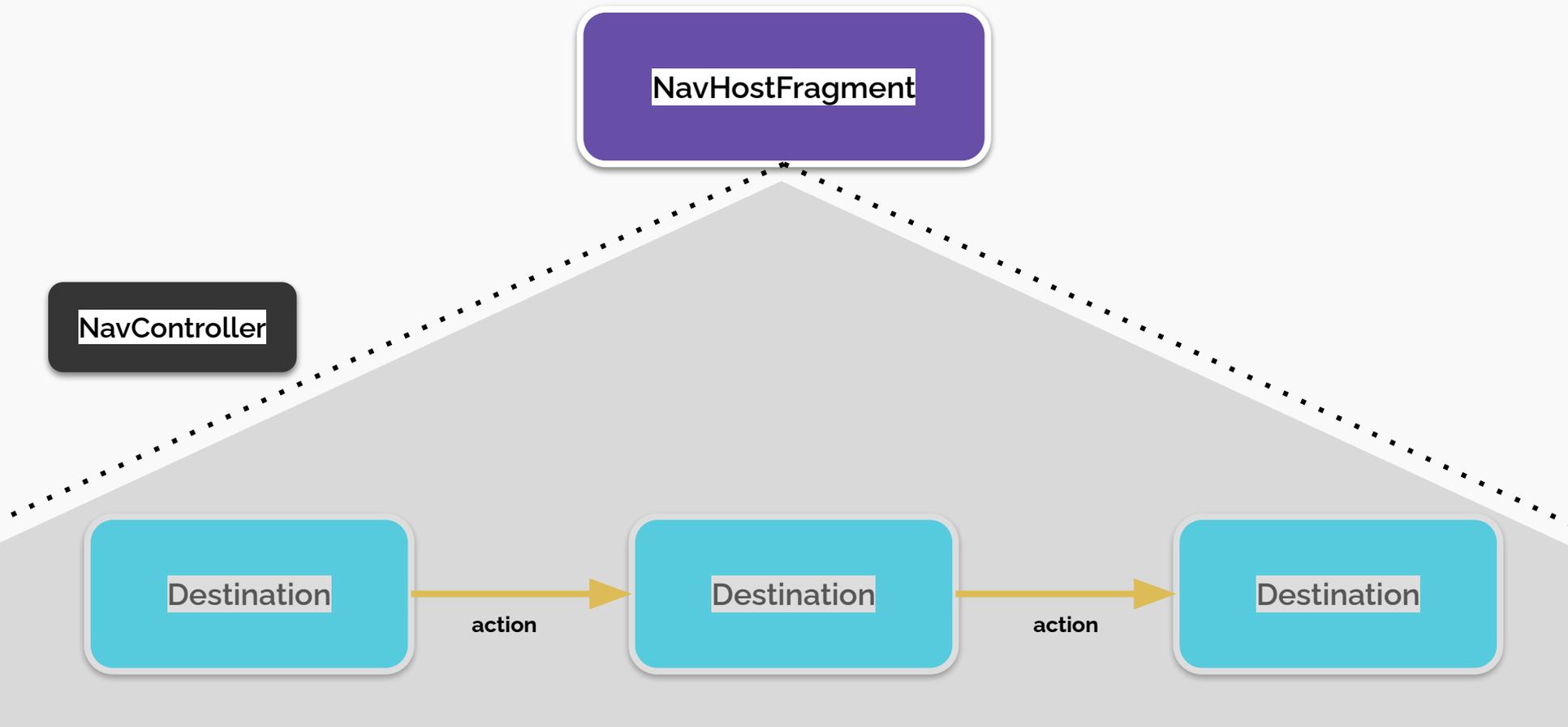
Destination

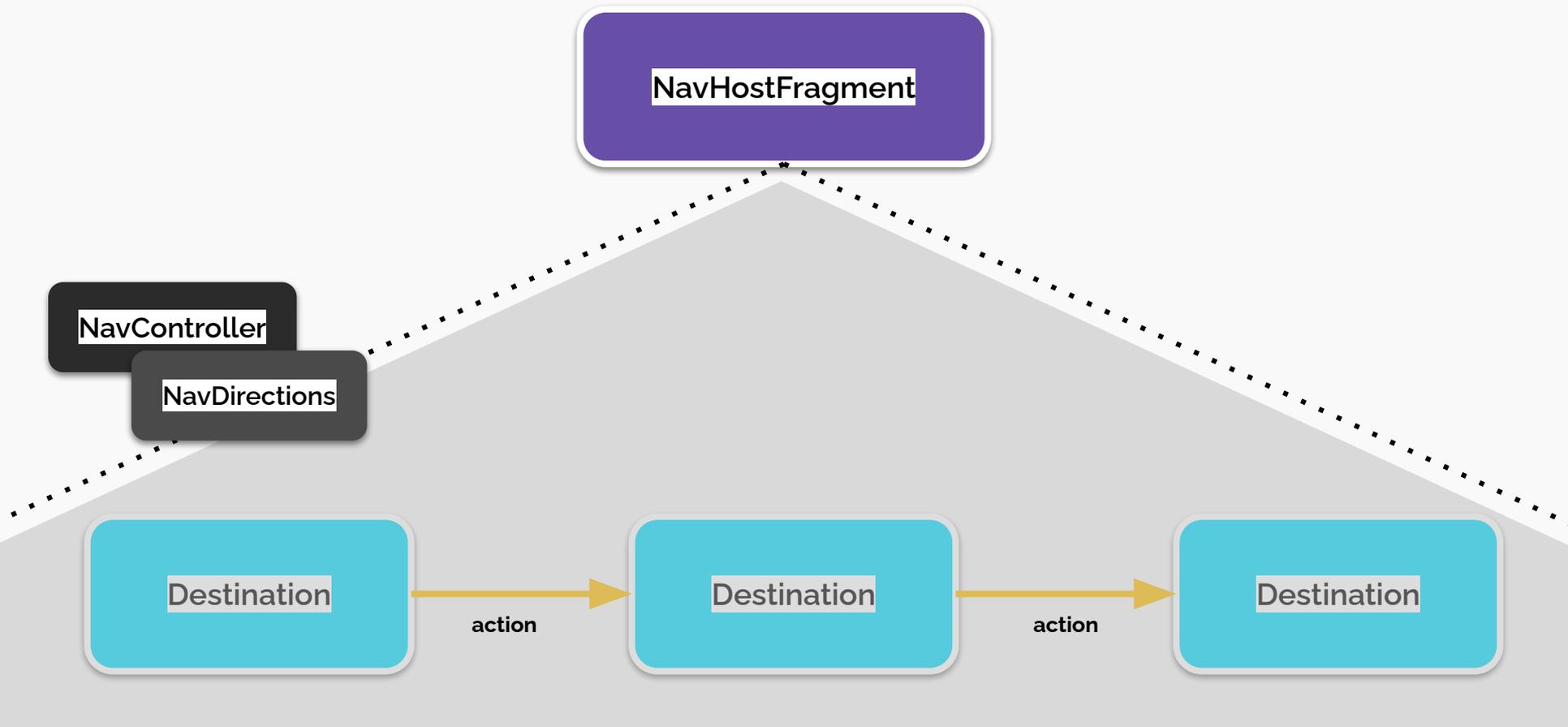
action

Destination

action

Destination

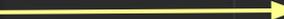




```
btn_login.setOnClickListener { it: View!  
    findNavController().navigate(  
        LoginFragmentDirections.toHomeFragment()  
    )  
}
```

```
btn_register.setOnClickListener { it: View!  
    findNavController().navigate(  
        LoginFragmentDirections.toCreateAccountFragment()  
    )  
}
```

NavController



```
btn_login.setOnClickListener { it: View!  
    findNavController().navigate(  
        LoginFragmentDirections.toHomeFragment()  
    )  
}  
  
btn_register.setOnClickListener { it: View!  
    findNavController().navigate(  
        LoginFragmentDirections.toCreateAccountFragment()  
    )  
}
```

NavController

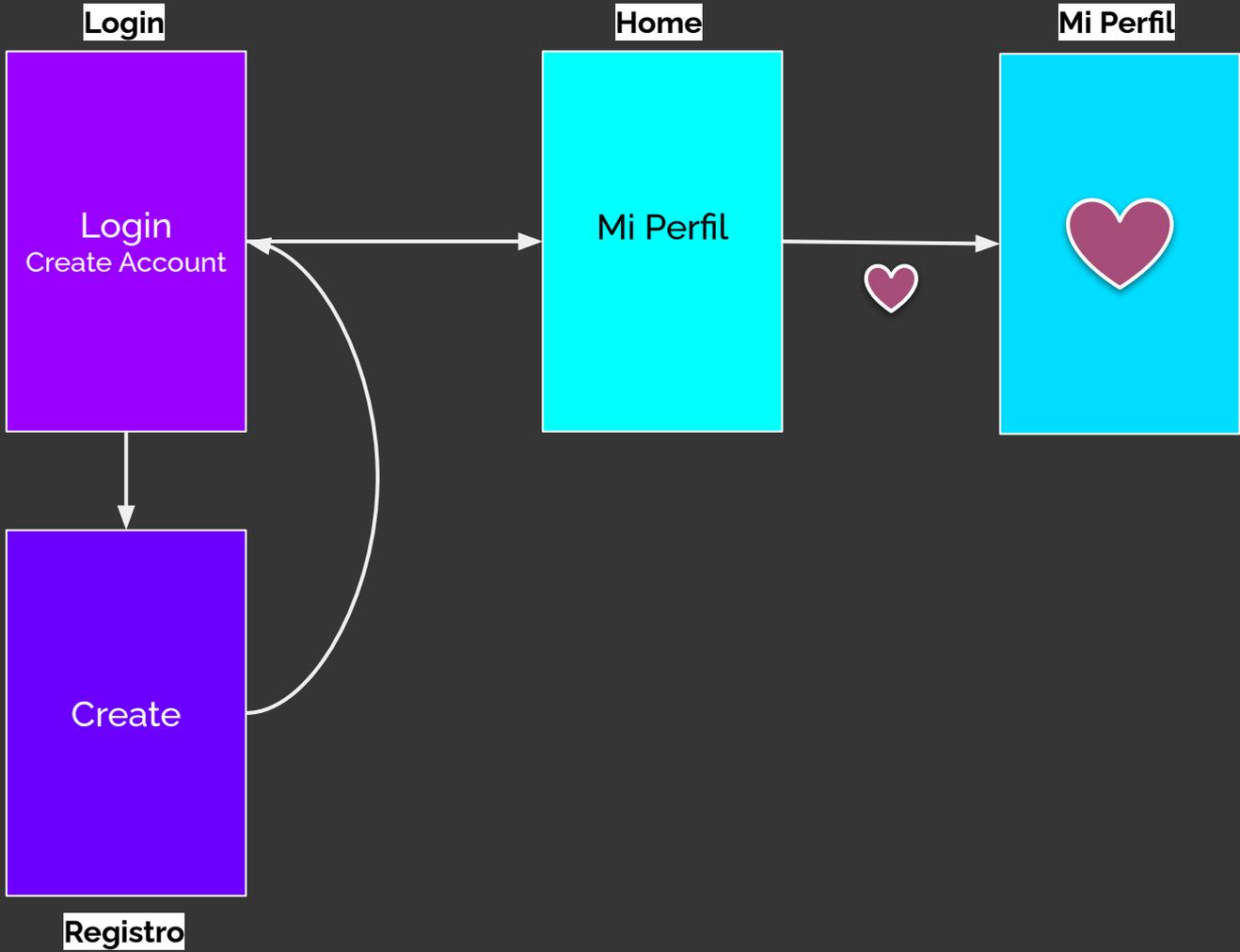
```
btn_login.setOnClickListener { it: View!  
    findNavController().navigate(  
        LoginFragmentDirections.toHomeFragment()  
    )  
}
```

NavDirections

```
btn_register.setOnClickListener { it: View!  
    findNavController().navigate(  
        LoginFragmentDirections.toCreateAccountFragment()  
    )  
}
```

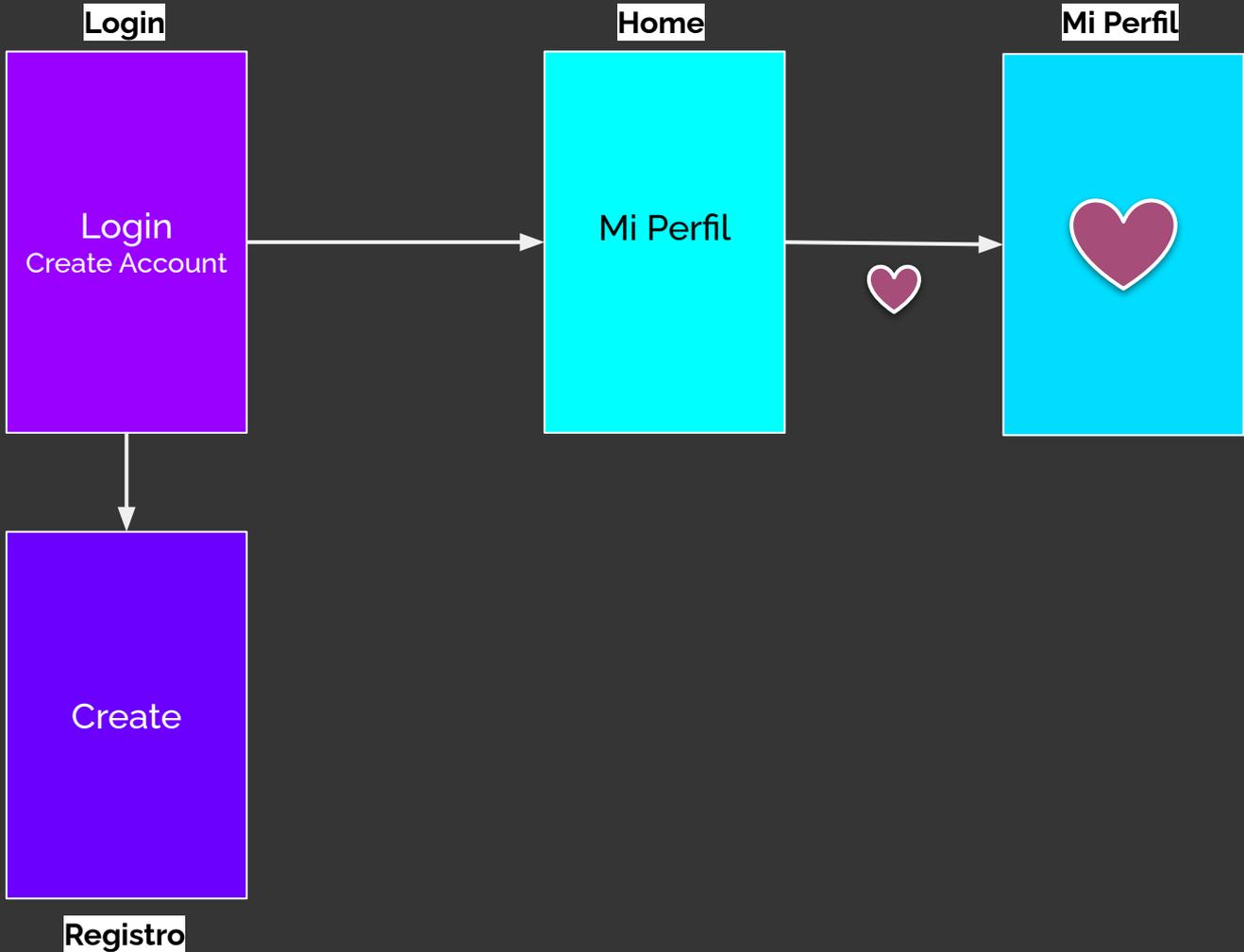
Example

—



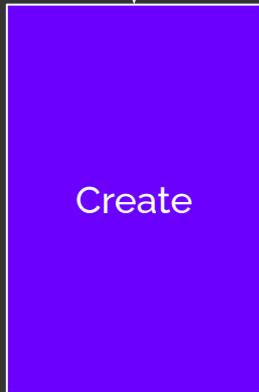
Stack

—

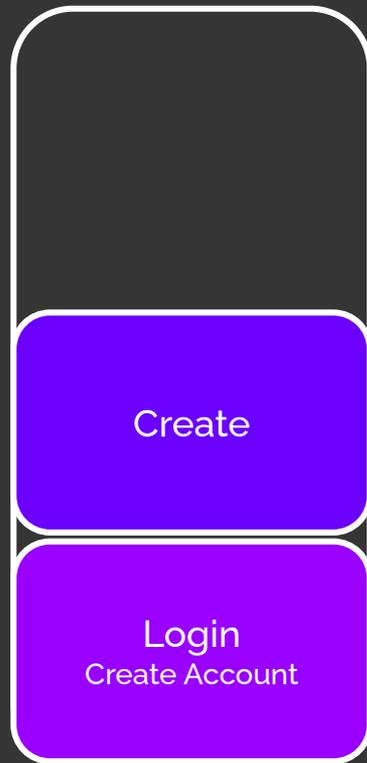


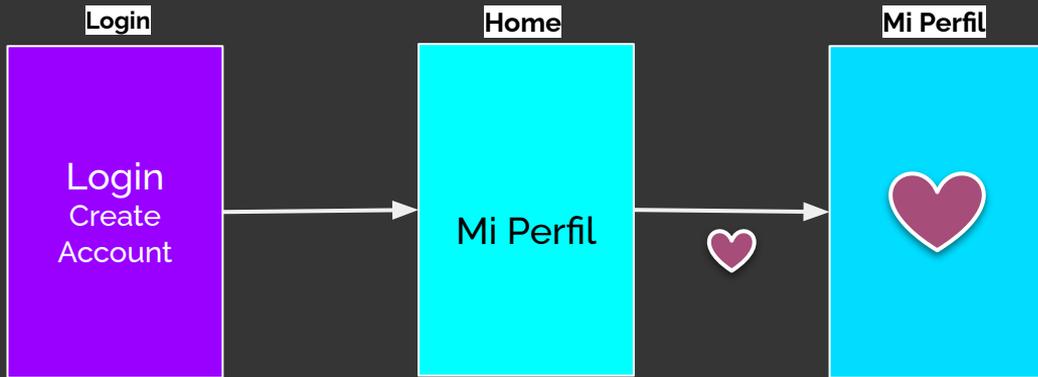
—

Login



Registro





Animations

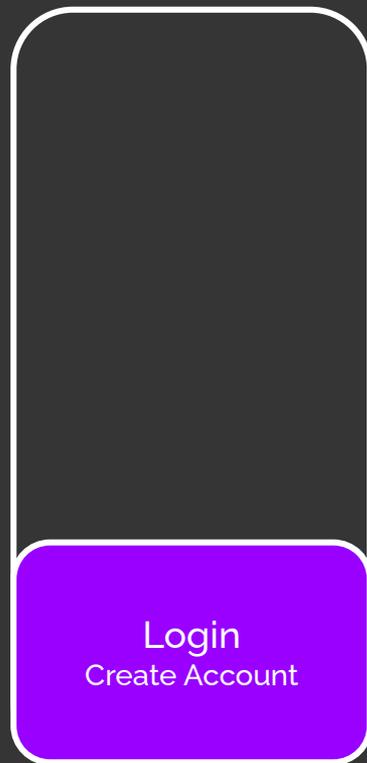
—

Login

Login
Create Account

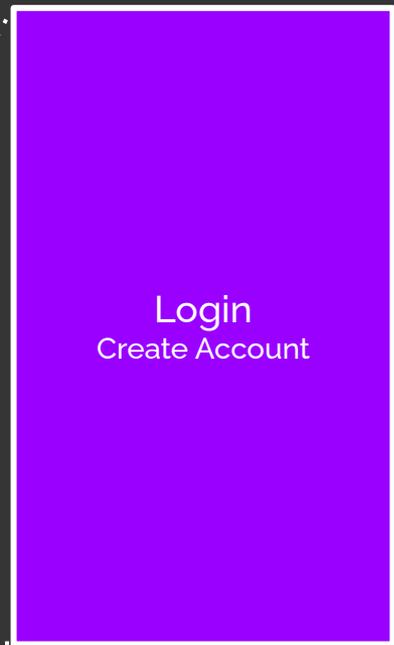
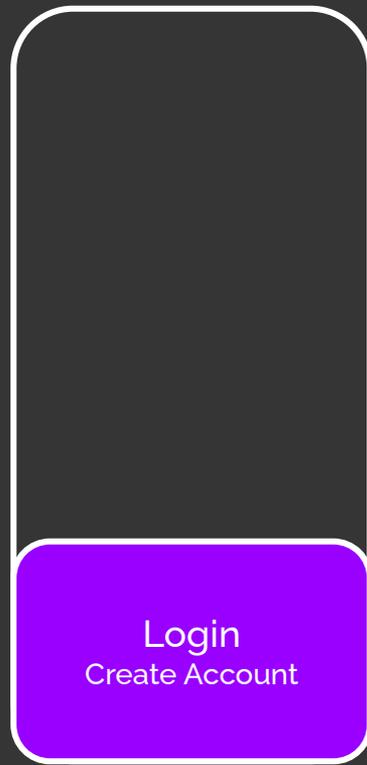
—

Login



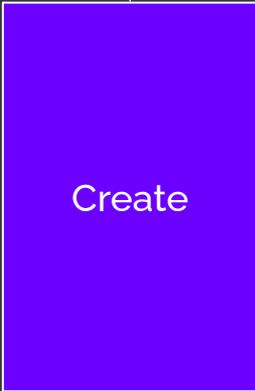
—

Login



—

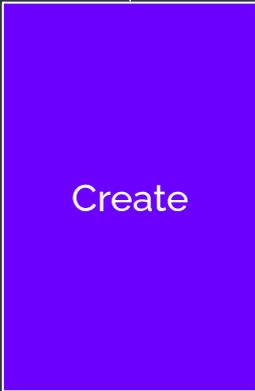
Login



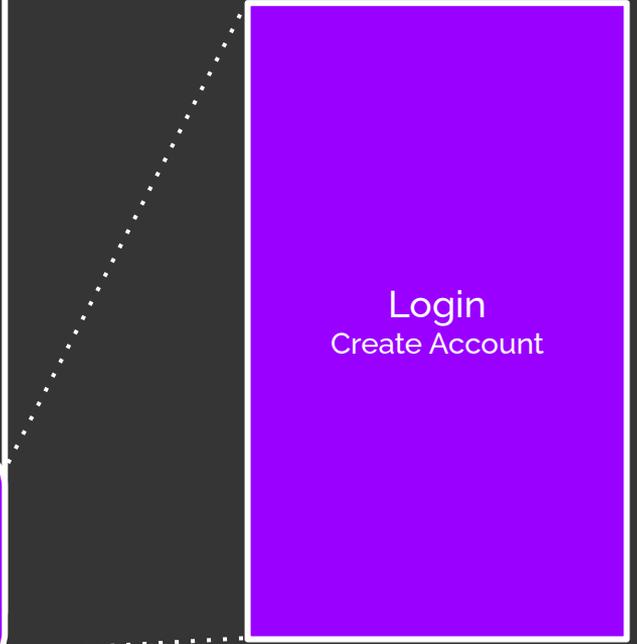
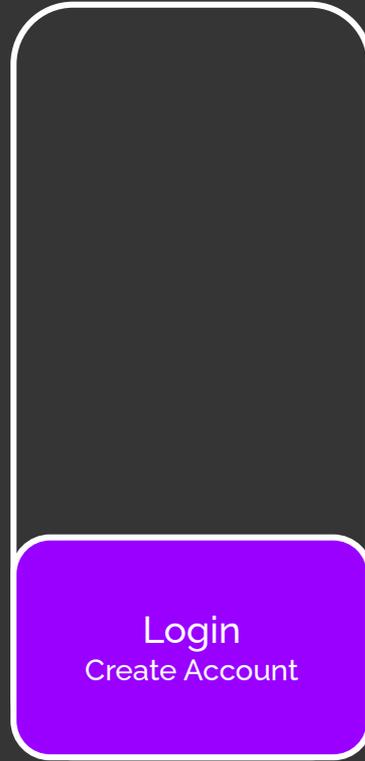
Registro

—

Login

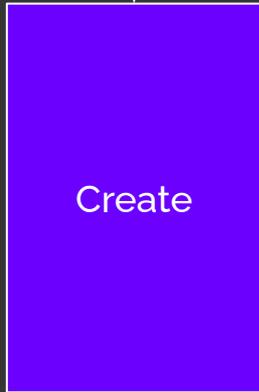


Registro

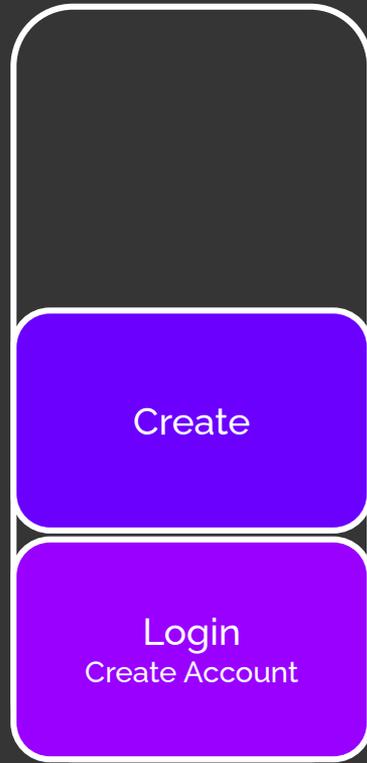


—

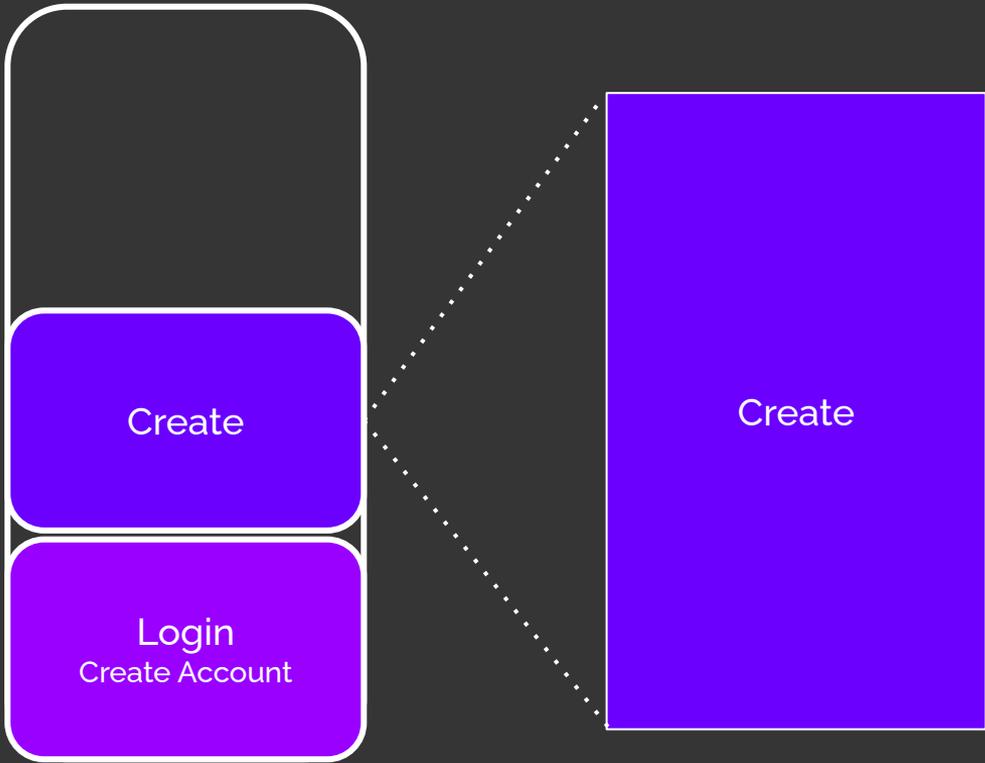
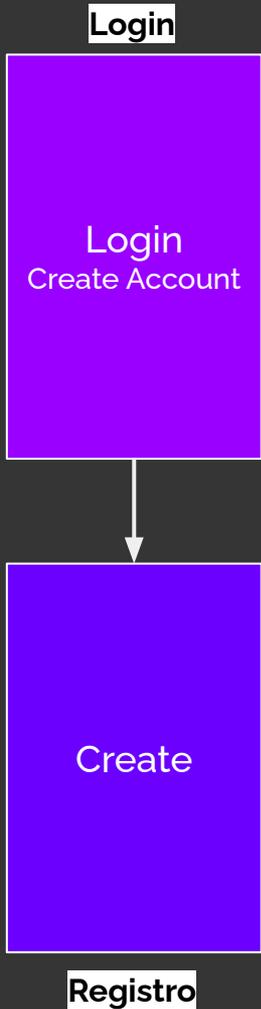
Login



Registro

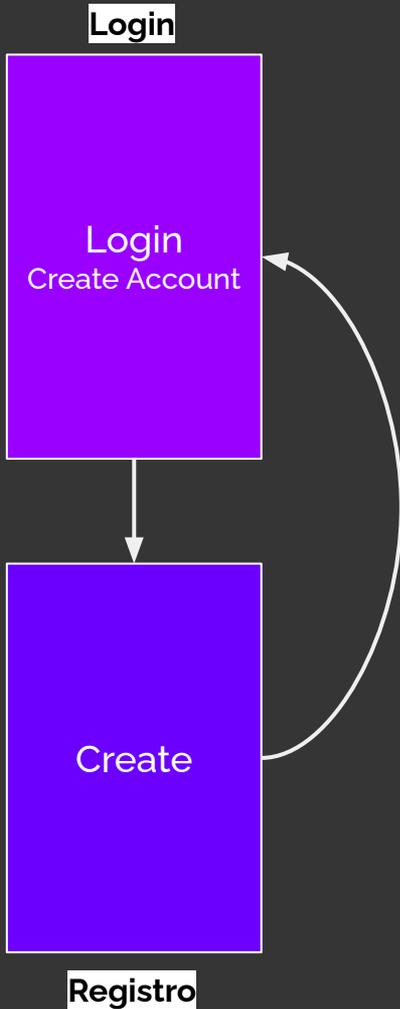


—



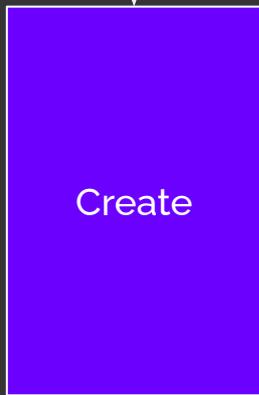
BackStack

—

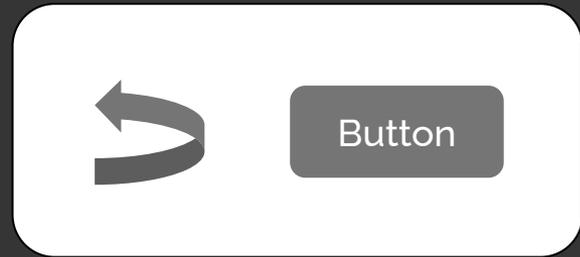


—

Login

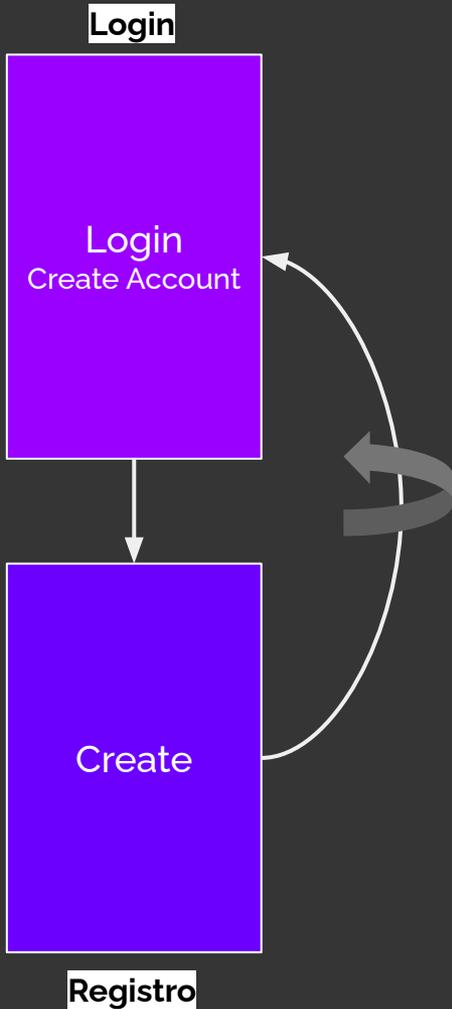


Registro

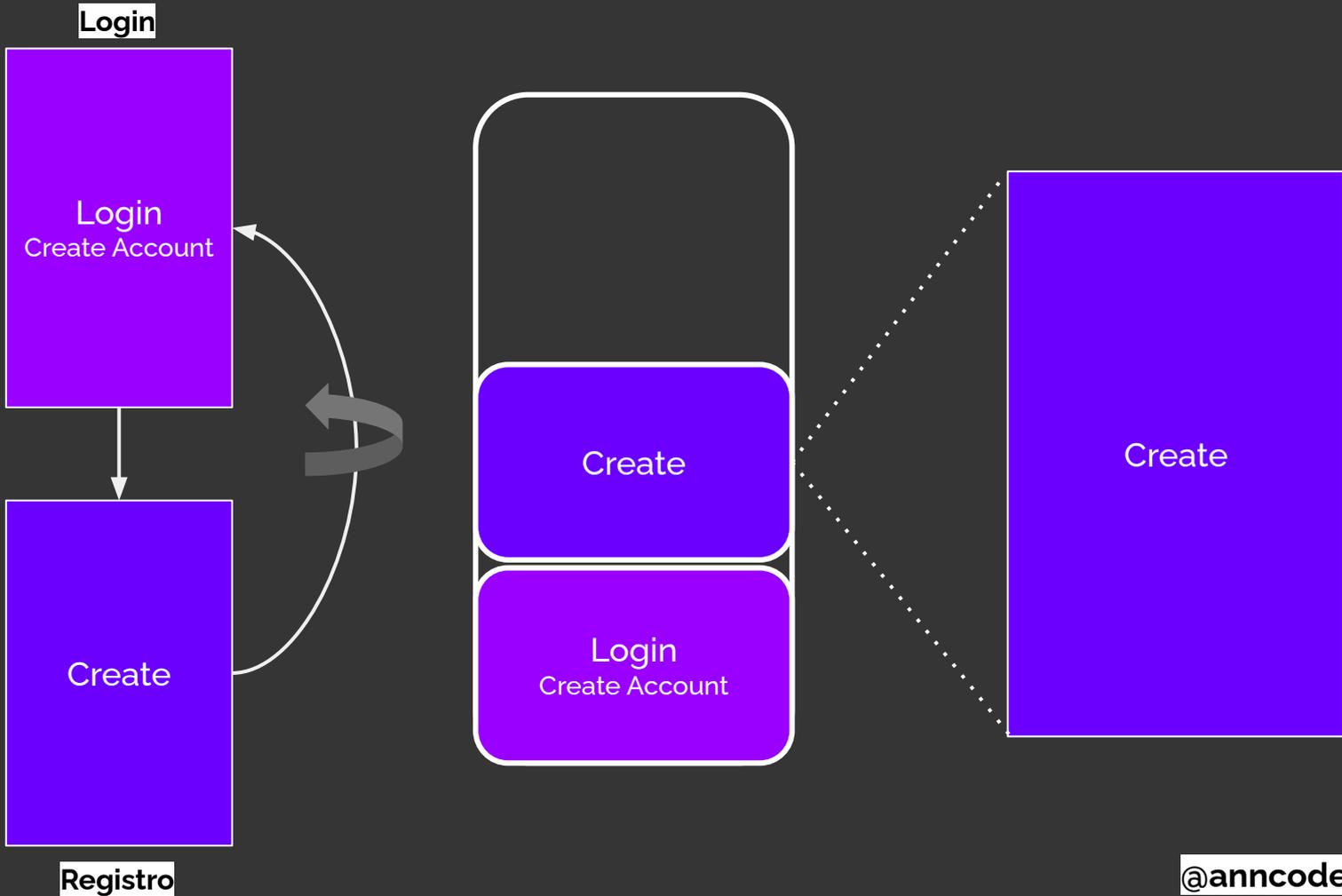


Atrás

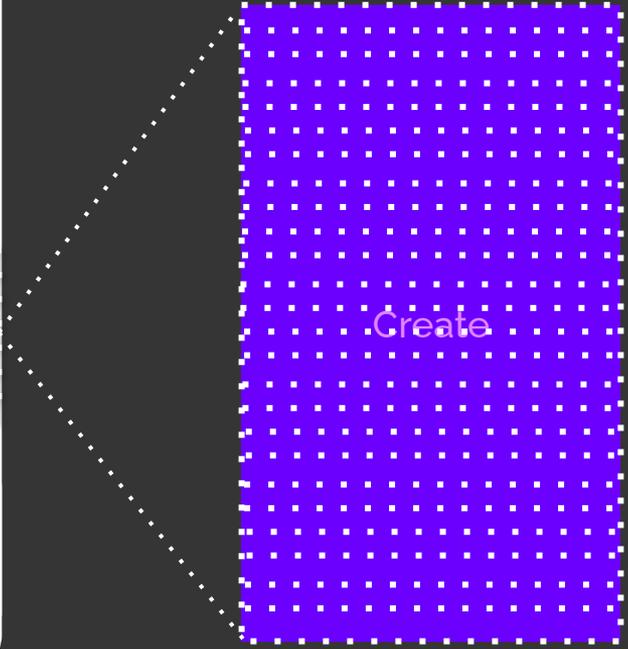
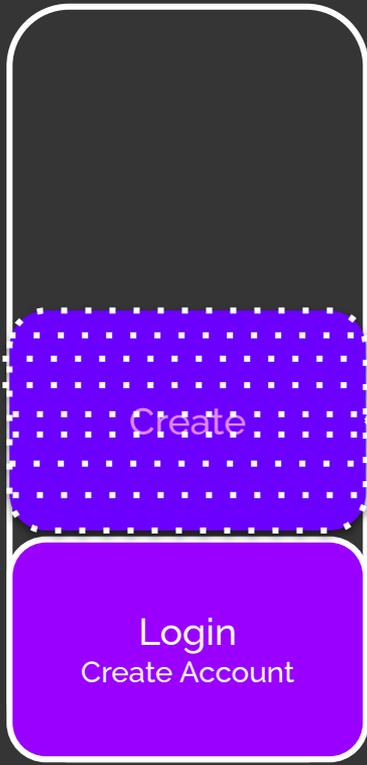
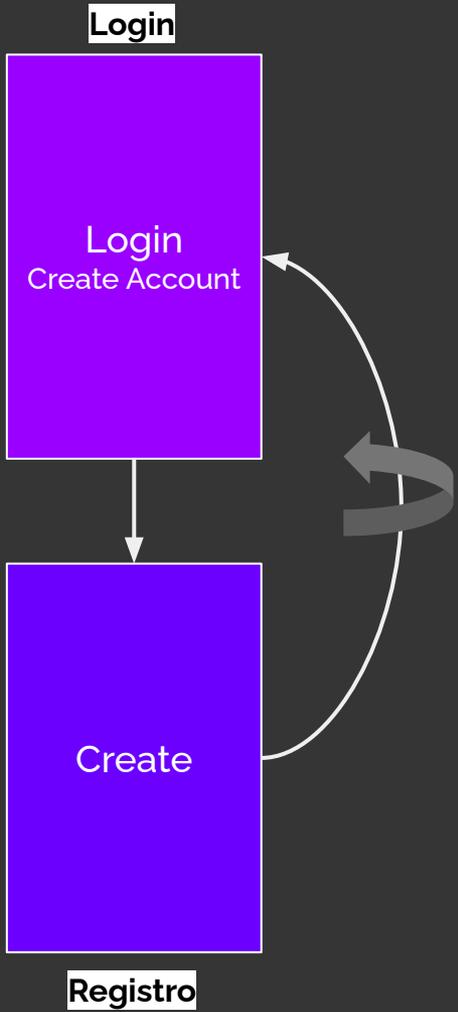
—



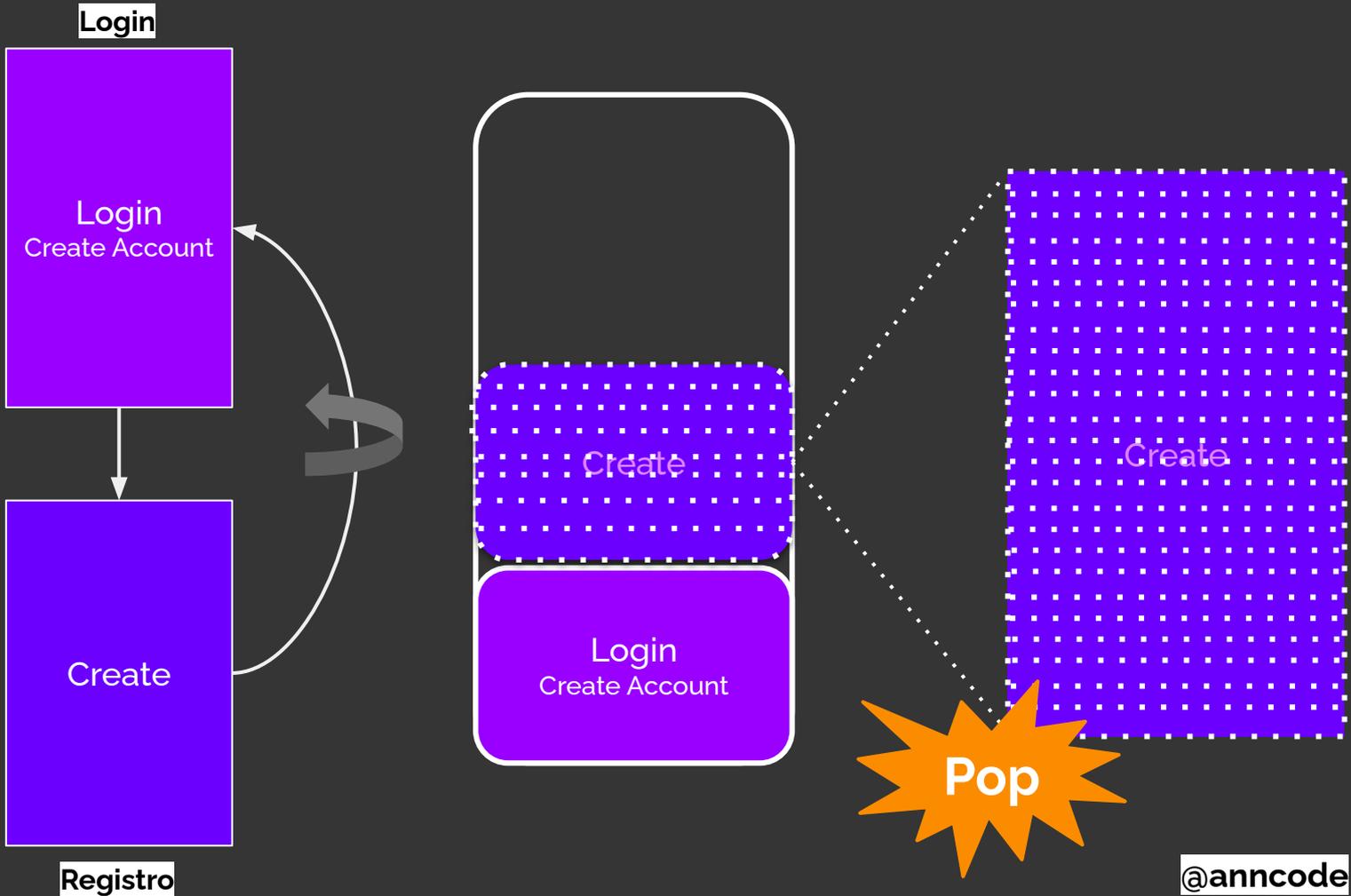
—



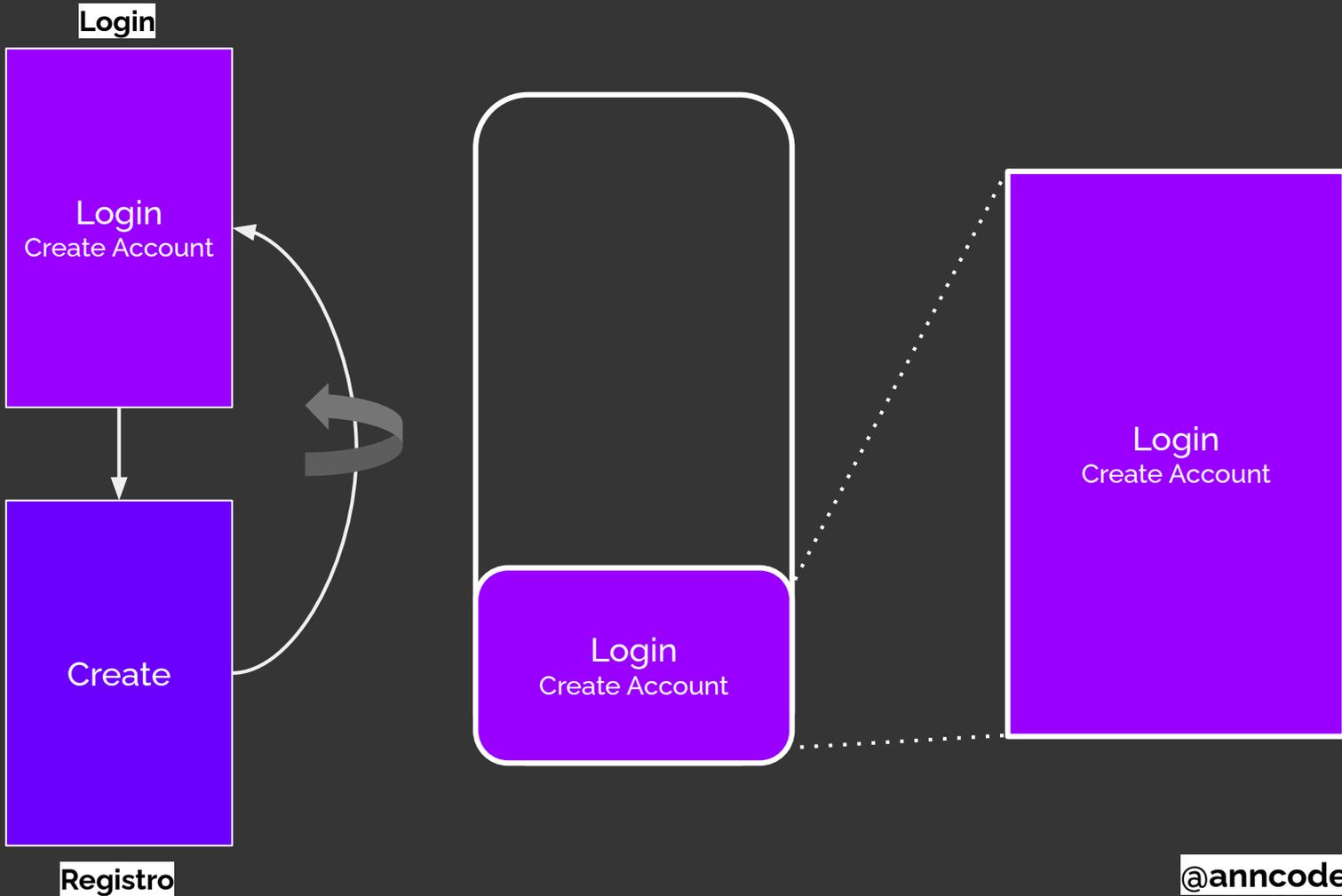
—



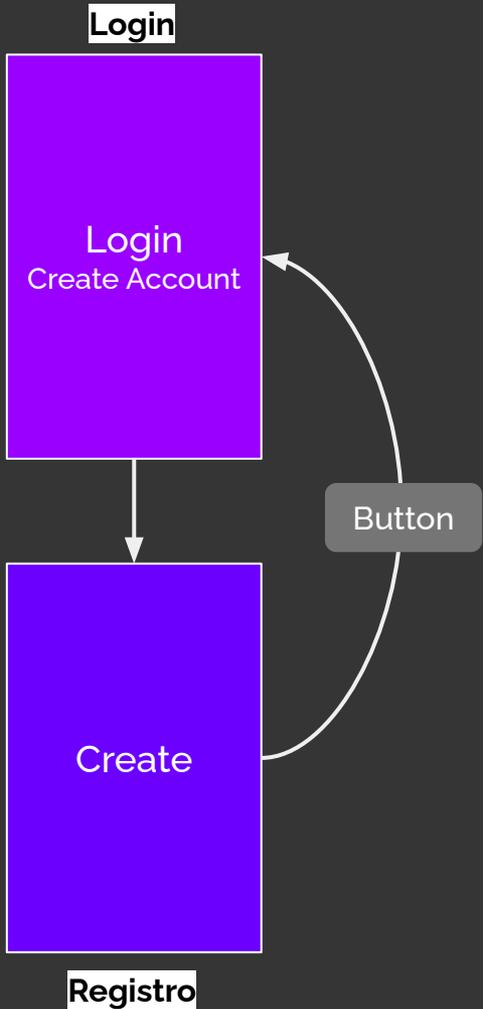
—



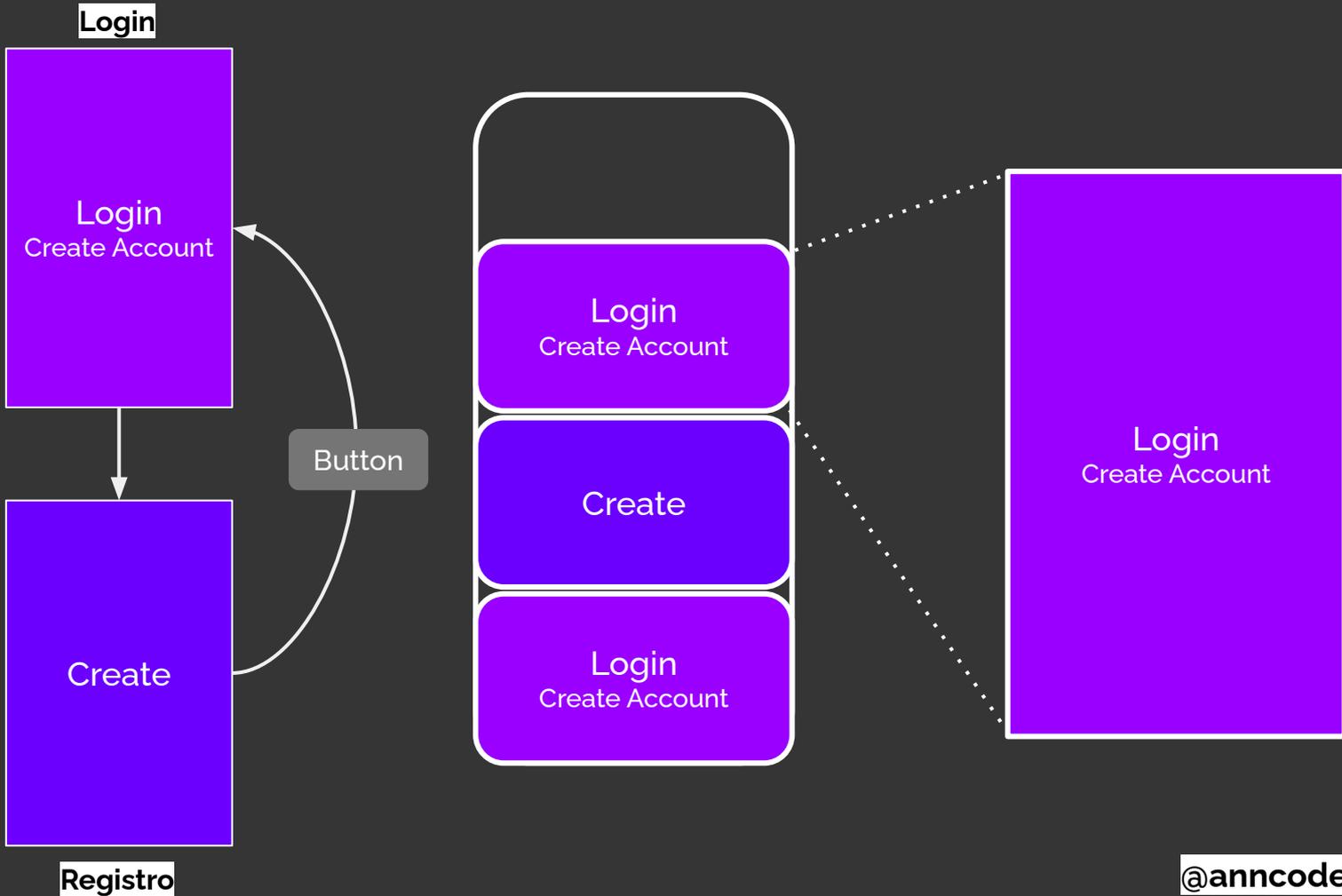
—



—

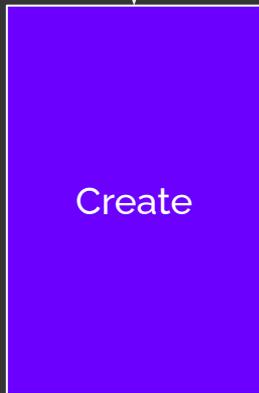


—



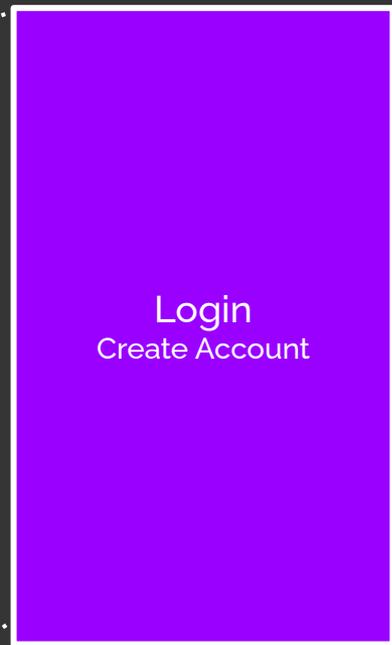
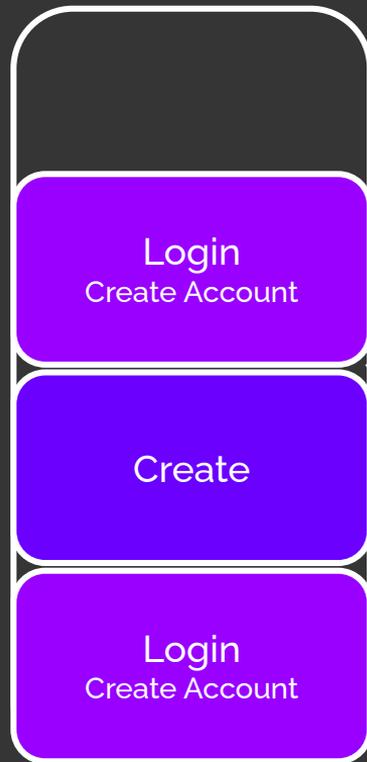
—

Login

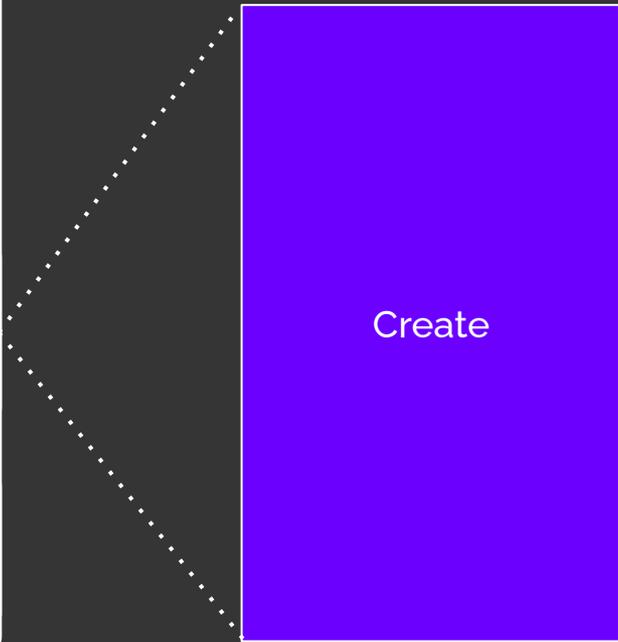
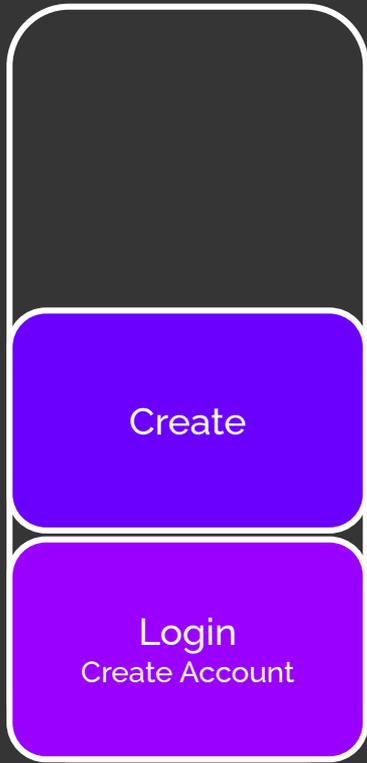
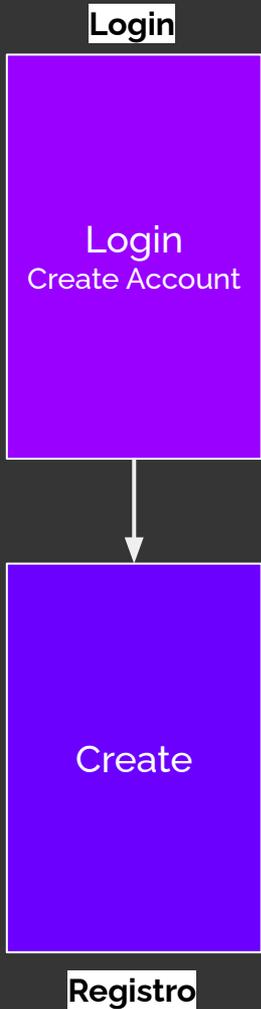


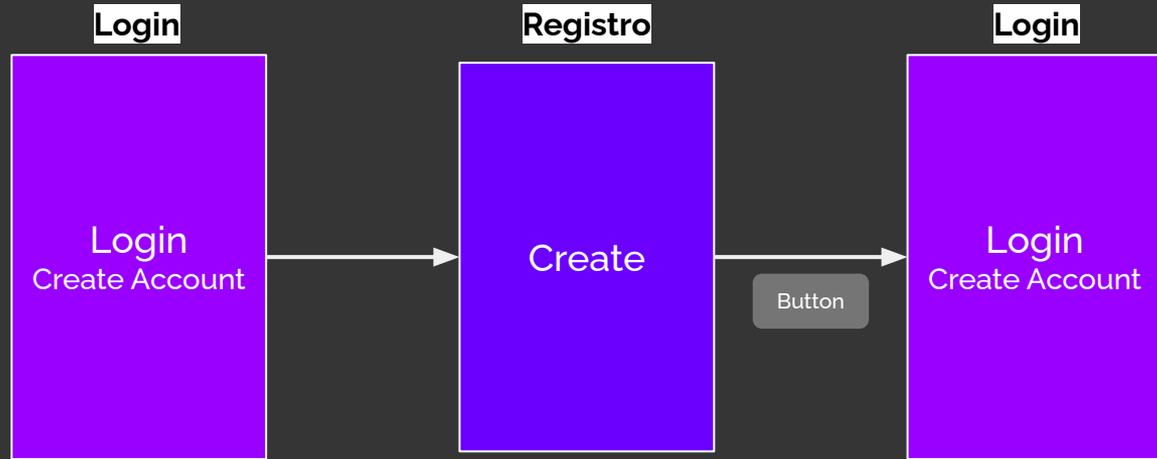
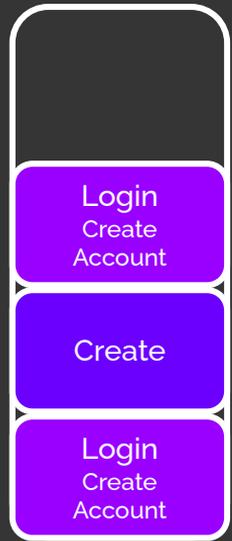
Registro

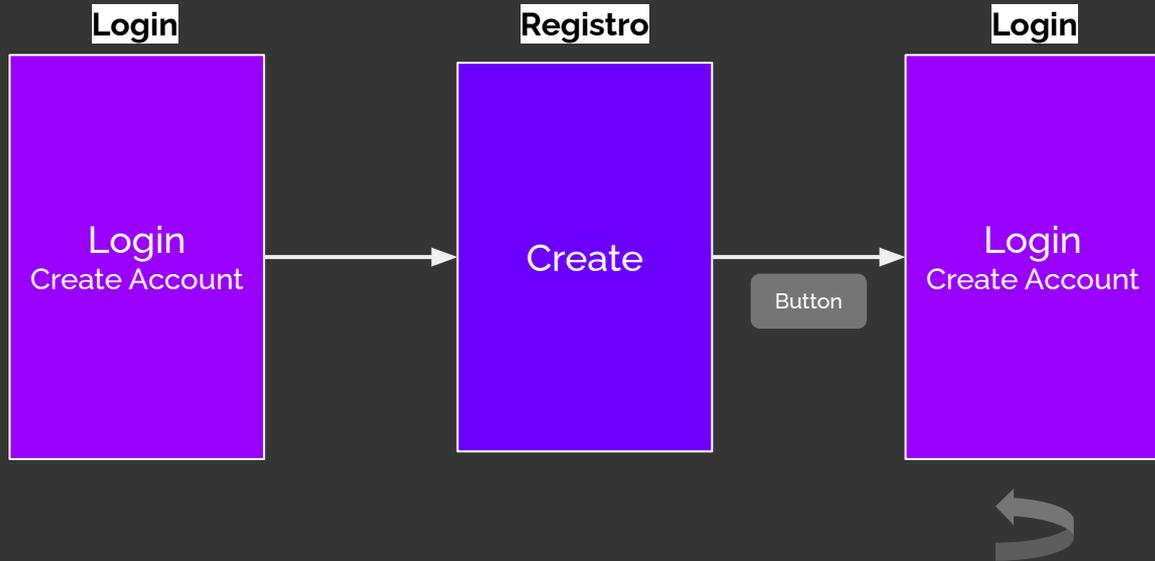
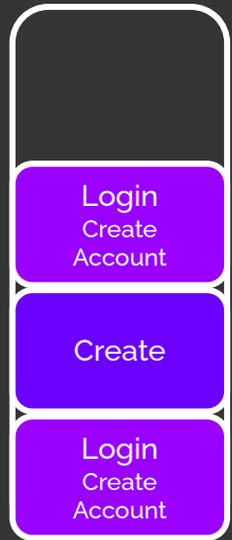
Button



—



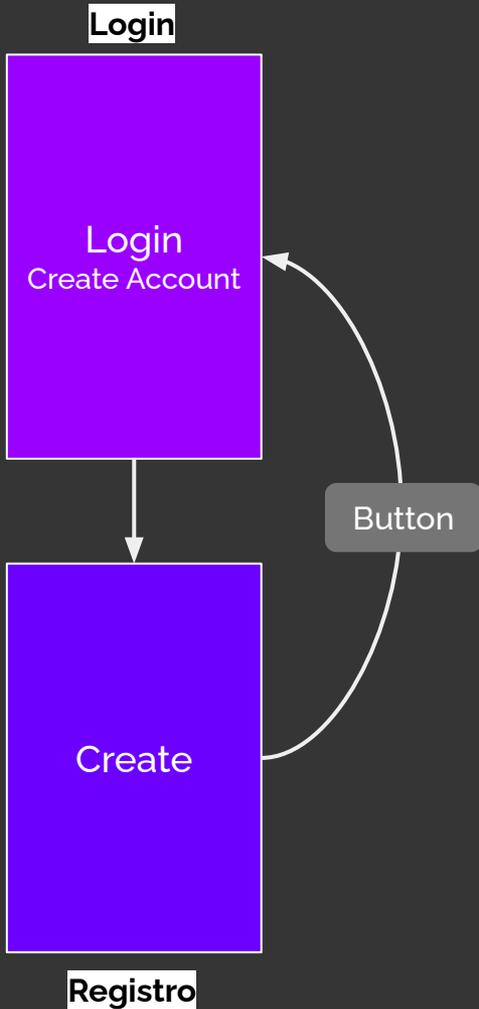




popUpTo

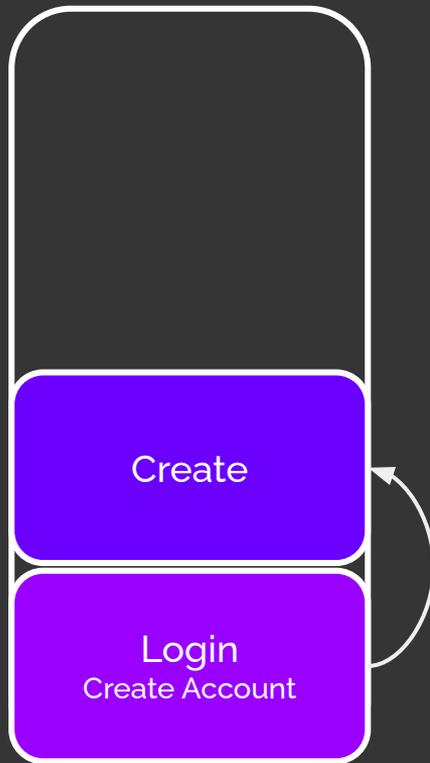
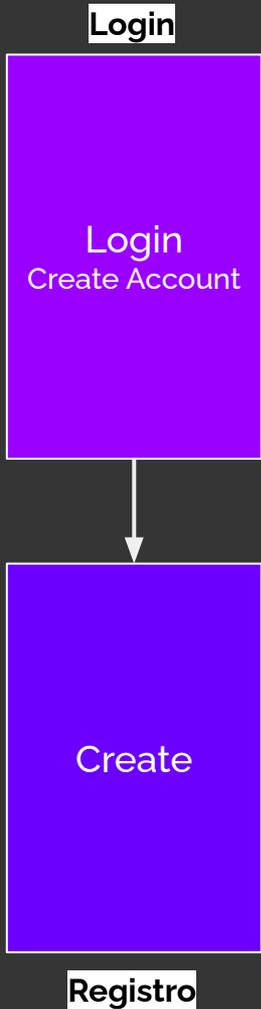
popUpTo

—



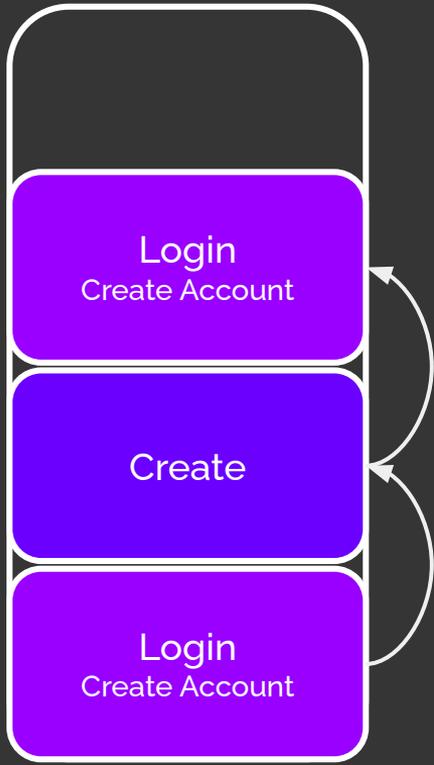
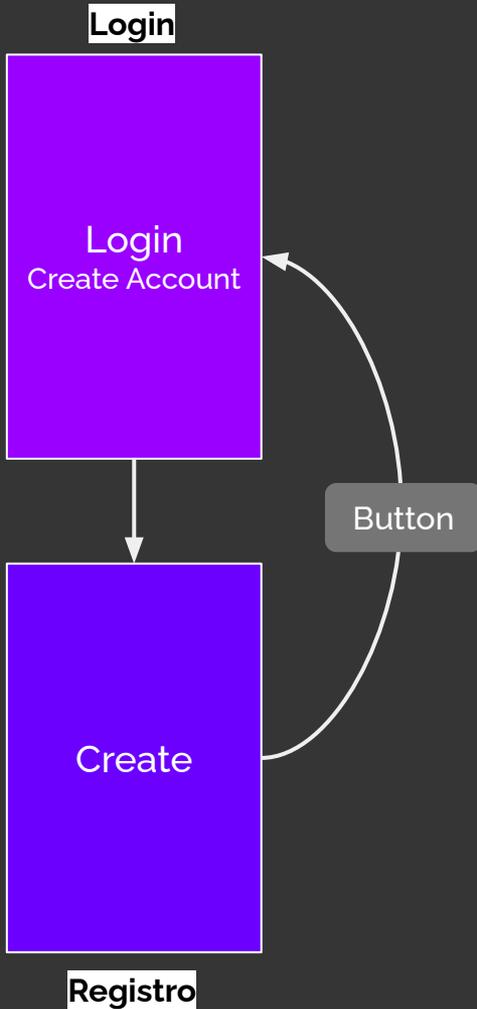
popUpTo

—



action

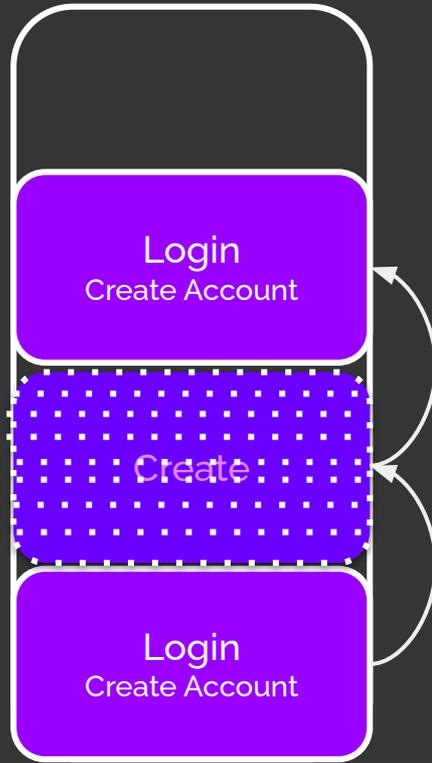
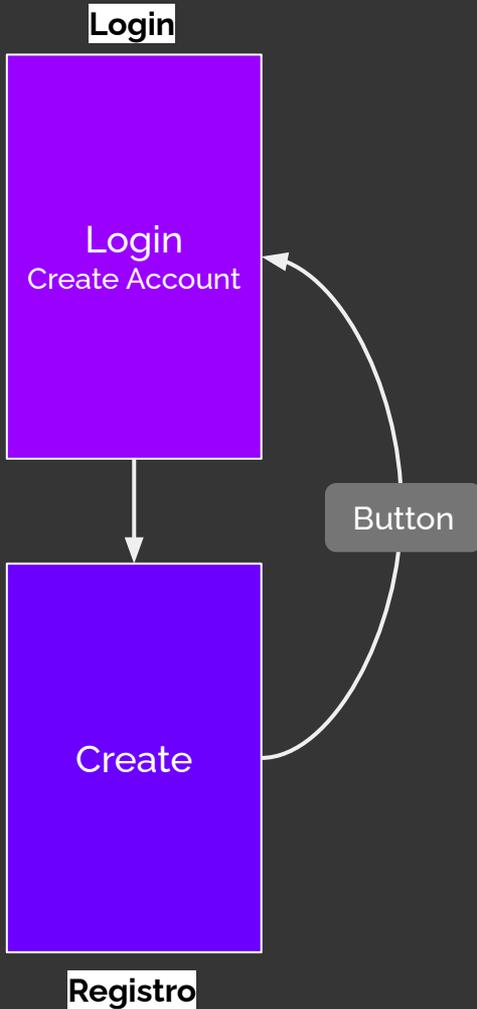
—



popUpTo

action

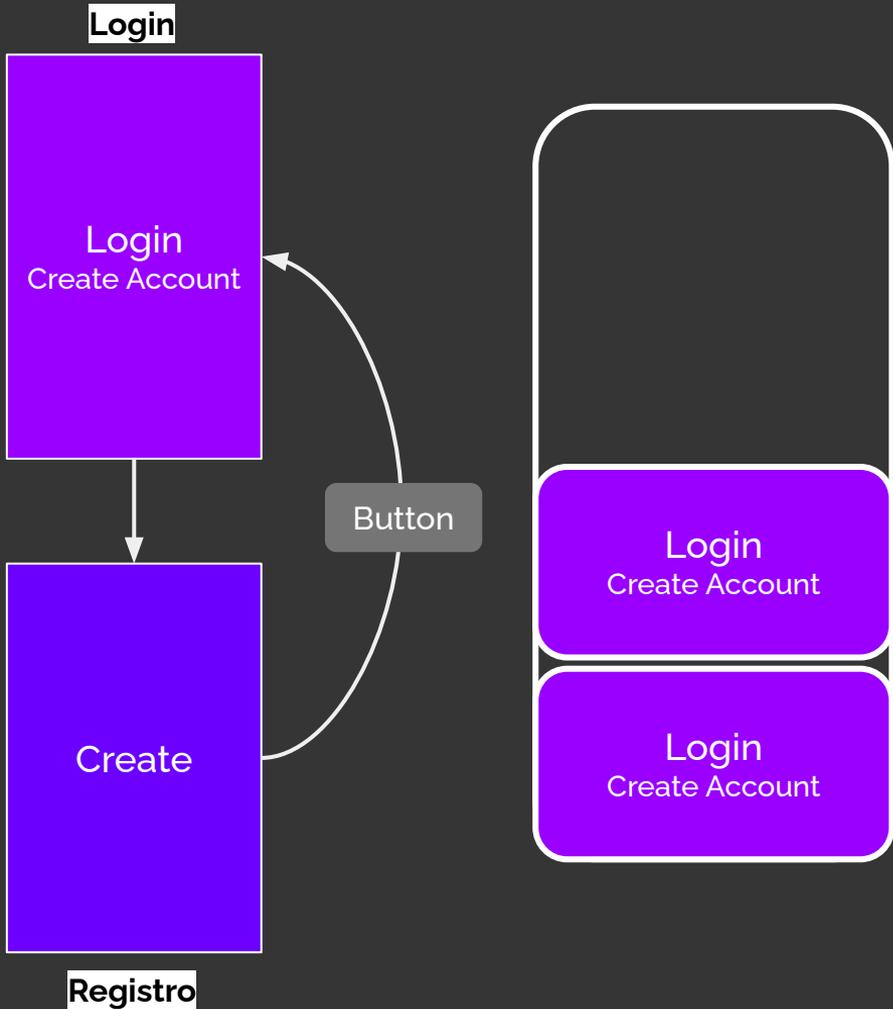
—



popUpTo

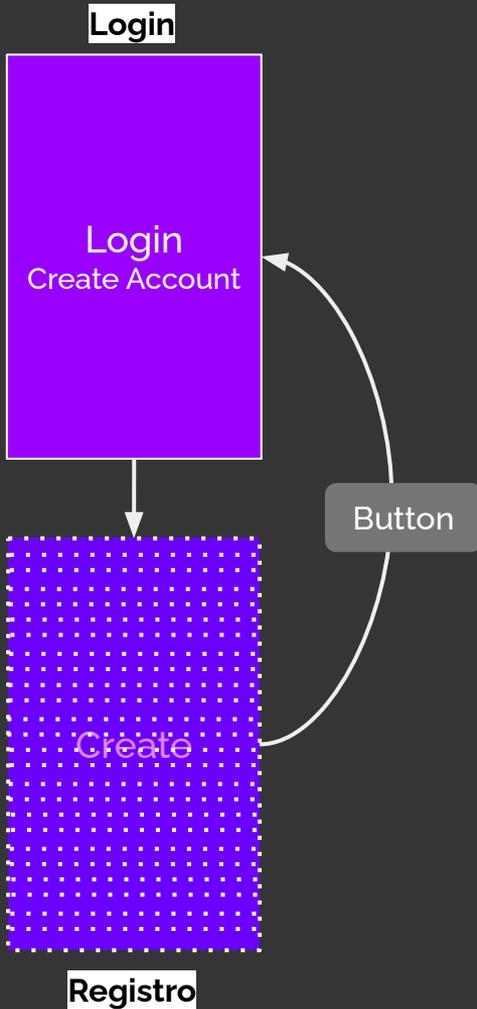
action

—



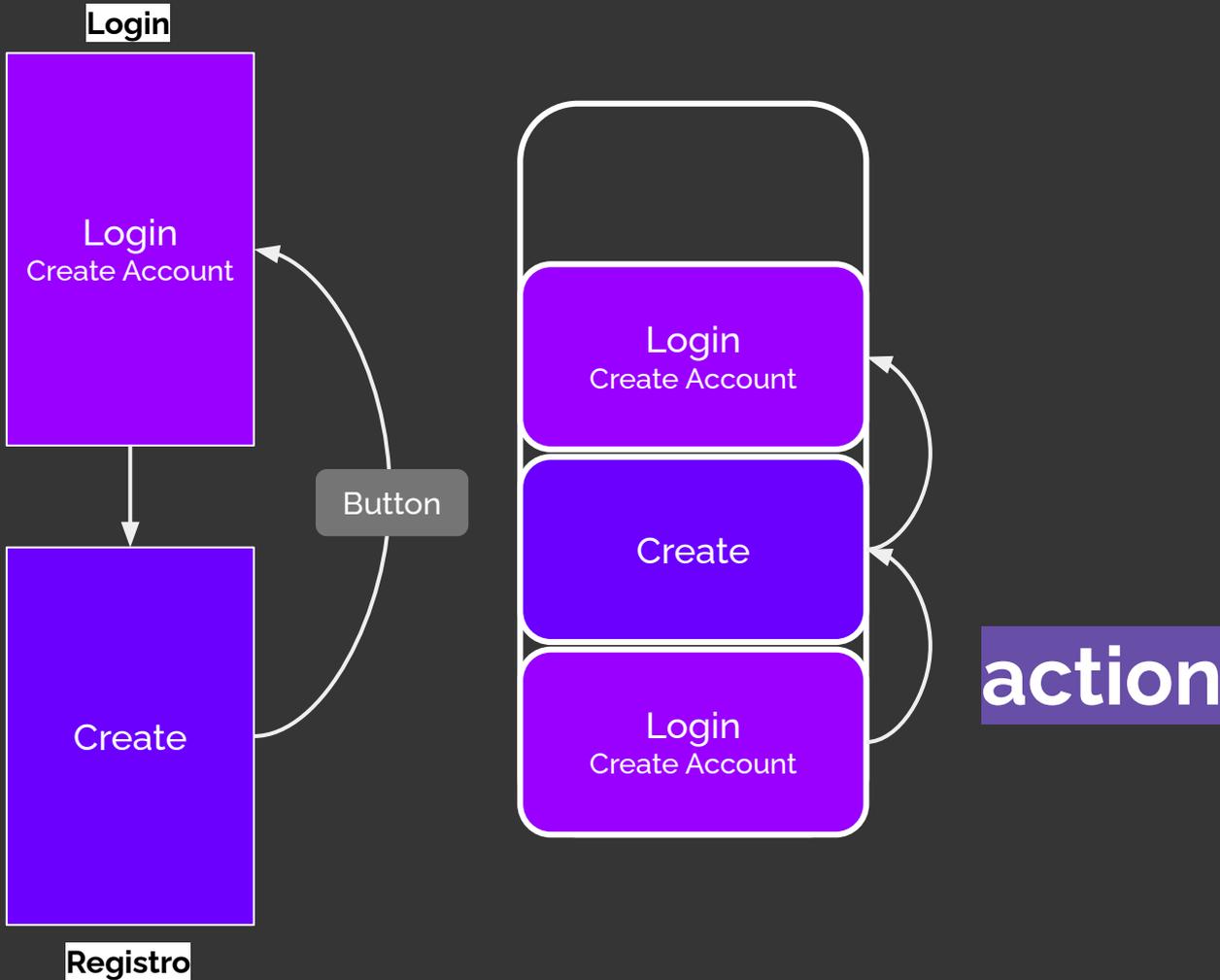
popUpToInclusive

—

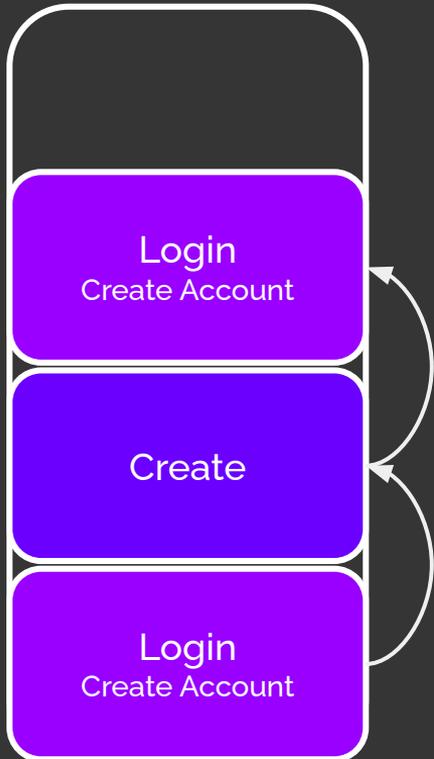
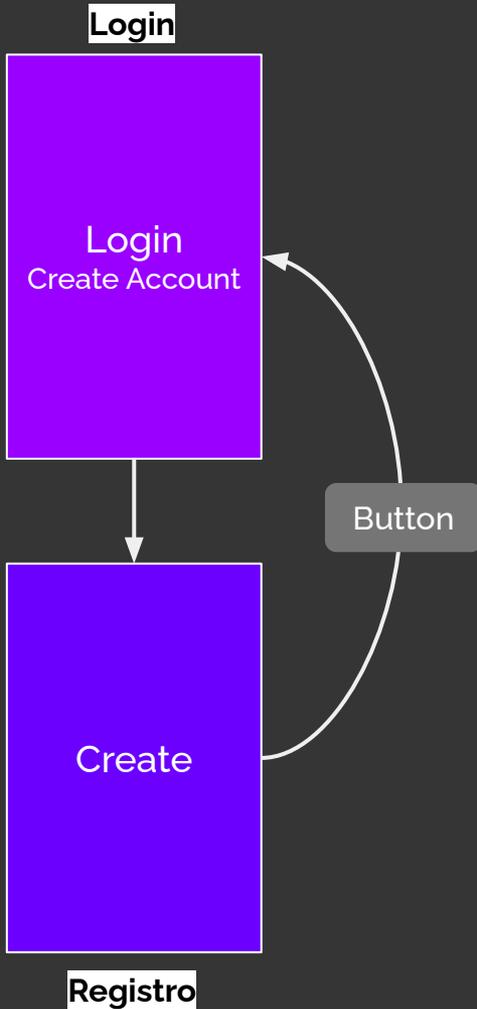


popUpTo
popUpToInclusive

—



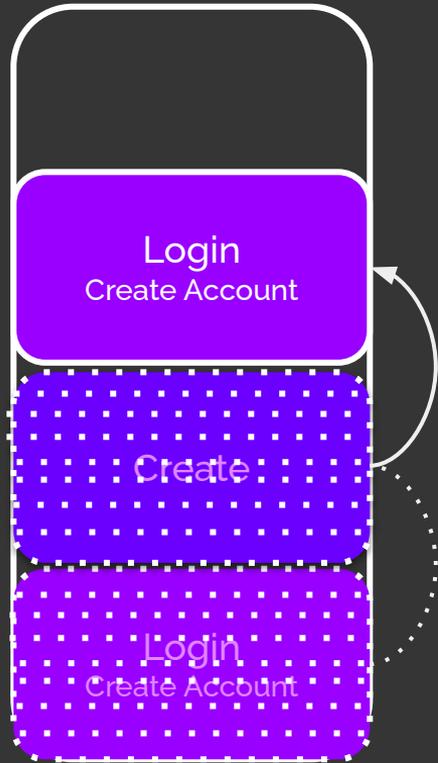
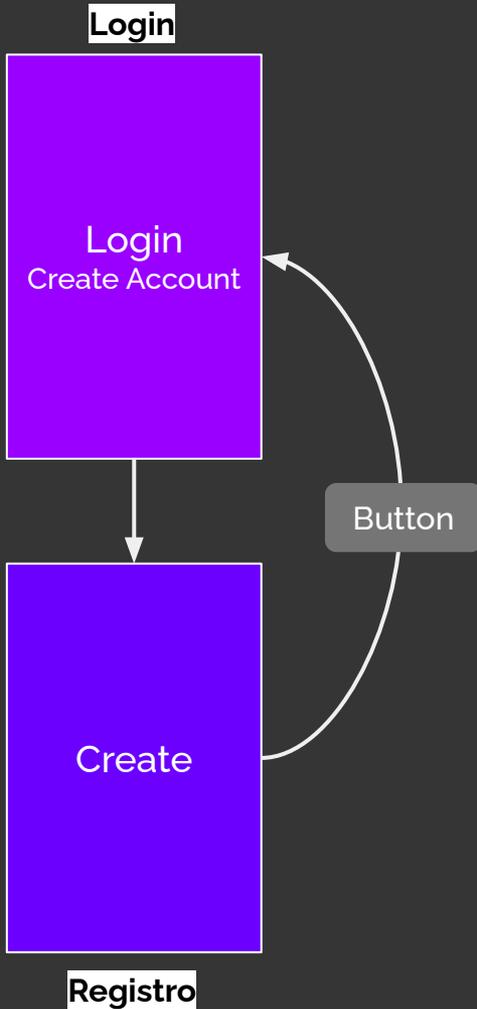
—



popUpTo
popUpToInclusive

action

—

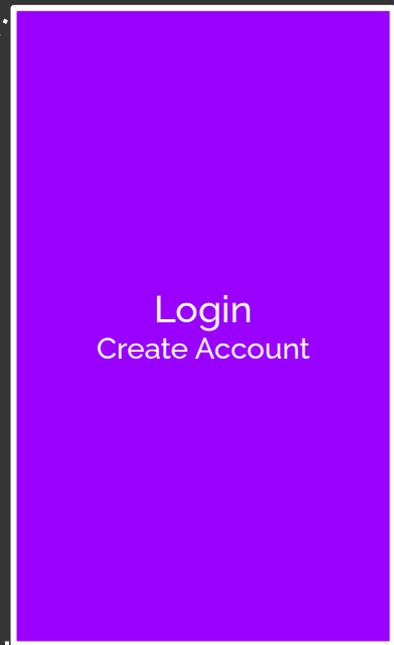
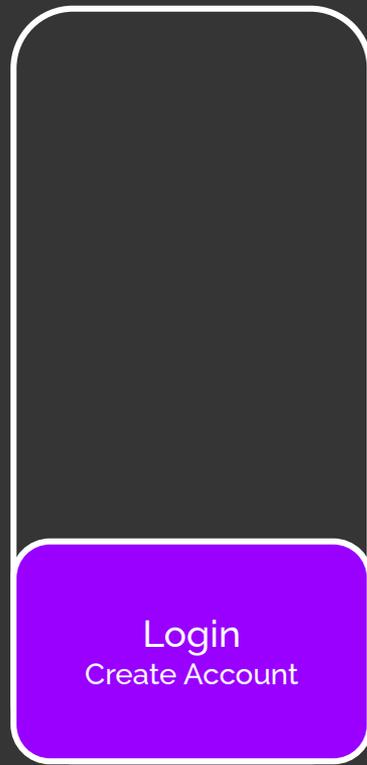


popUpTo
popUpToInclusive

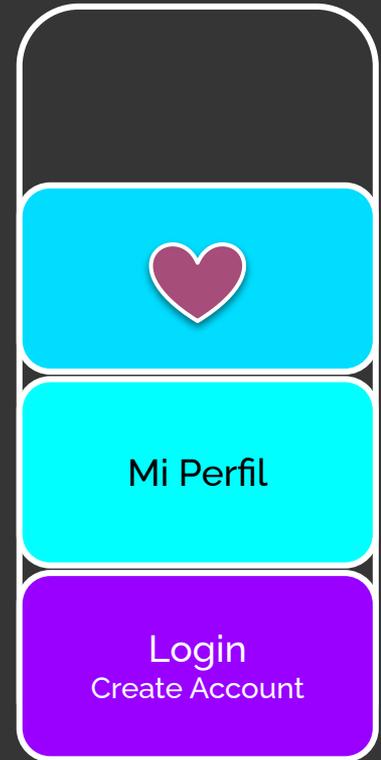
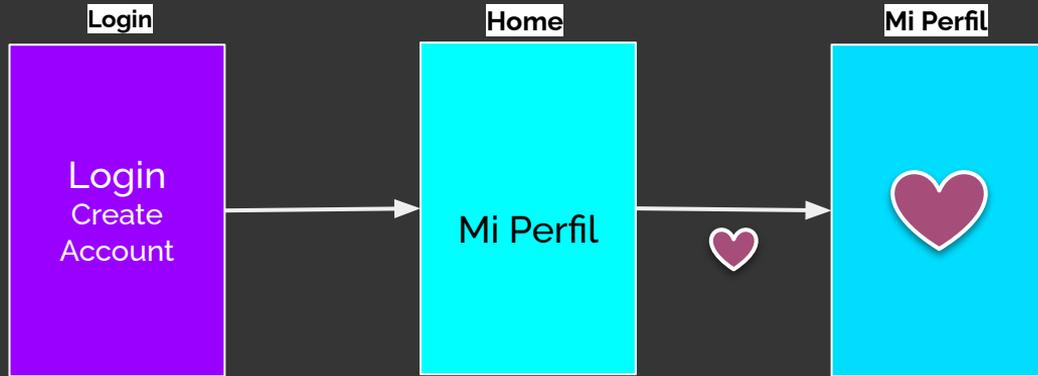
action

—

Login

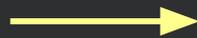


Arregla esto suponiendo que ya hiciste login



args

Argument



Attributes

myProfileFragment fragment

id myProfileFragment

label MyProfileFragment

name MyProfileFragment (com.anncode.androidnavic

Arguments + -

user: com.anncode.androidnavigationcomponent.User? (@null)

Actions + -

Deep Links + -

Nothing to show

myProfileFragment

My Profile

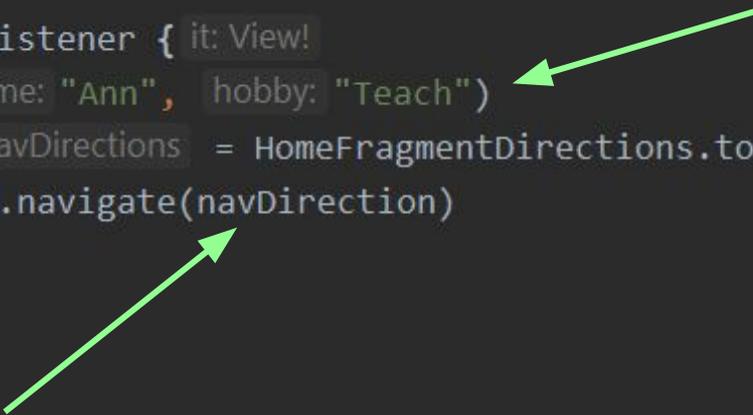
Argument

```
<fragment
  android:id="@+id/myProfileFragment"
  android:name="com.anncode.androidnavigationcomponent.MyProfileFragment"
  android:label="MyProfileFragment"
  tools:layout="@layout/fragment_my_profile"
  >
  <argument
    android:name="user"
    app:argType="com.anncode.androidnavigationcomponent.User"
    app:nullable="true"
    android:defaultValue="@null" />
</fragment>
```

```
btn_profile.setOnClickListener { it: View!  
    val user = User( name: "Ann", hobby: "Teach")  
    val navDirection : NavDirections = HomeFragmentDirections.toMyProfileFragment(user)  
    findNavController().navigate(navDirection)  
}
```

Enviar Argumento

```
btn_profile.setOnClickListener { it: View!  
    val user = User( name: "Ann", hobby: "Teach")  
    val navDirection : NavDirections = HomeFragmentDirections.toMyProfileFragment(user)  
    findNavController().navigate(navDirection)  
}
```



Enviar Argumento

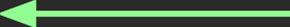
```
class MyProfileFragment : Fragment() {  
  
    private val args: MyProfileFragmentArgs? by navArgs()  
  
    override fun onCreateView(view: View, savedInstanceState: Bundle?) {  
        super.onCreateView(view, savedInstanceState)  
  
        args?.user
```

Recibir Argumento



```
class MyProfileFragment : Fragment() {
```

```
    private val args: MyProfileFragmentArgs? by navArgs()
```



```
    override fun onCreateView(view: View, savedInstanceState: Bundle?) {  
        super.onCreateView(view, savedInstanceState)
```

```
        args?.user
```



Recibir Argumento

Android Navigation Component

Puntos de inicio condicionados ✓

Android Navigation Component

Puntos de inicio condicionados ✓

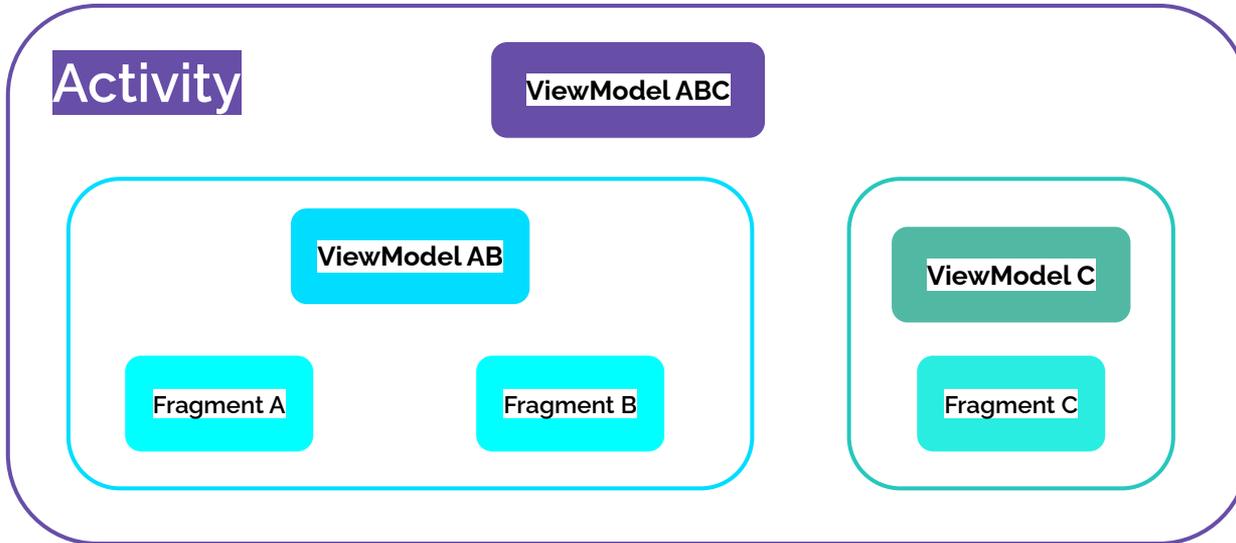
Grafos anidados - Nested Graph ✓

Android Navigation Component

Puntos de inicio condicionados ✓
Grafos anidados - Nested Graph ✓
ViewModels compartidos ✓

Android Navigation Component

ViewModels compartidos ✓



Android Navigation Component

Puntos de inicio condicionados ✓
Grafos anidados - Nested Graph ✓
ViewModels compartidos ✓

Android Navigation Component

- Puntos de inicio condicionados** ✓
- Grafos anidados - Nested Graph** ✓
- ViewModels compartidos** ✓
- Conectar con BottomNav** ✓

Android Navigation Component

- Puntos de inicio condicionados** ✓
- Grafos anidados - Nested Graph** ✓
- ViewModels compartidos** ✓
- Conectar con BottomNav** ✓
- Librería para hacer Test** ✓

Anahí Salgado

@anncode

