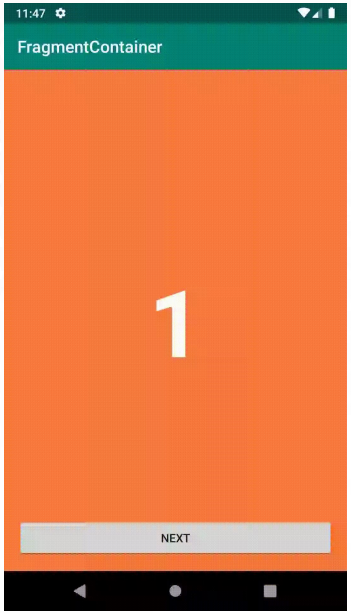
Aula 21
Fragments

DigitalHouse >
Coding School

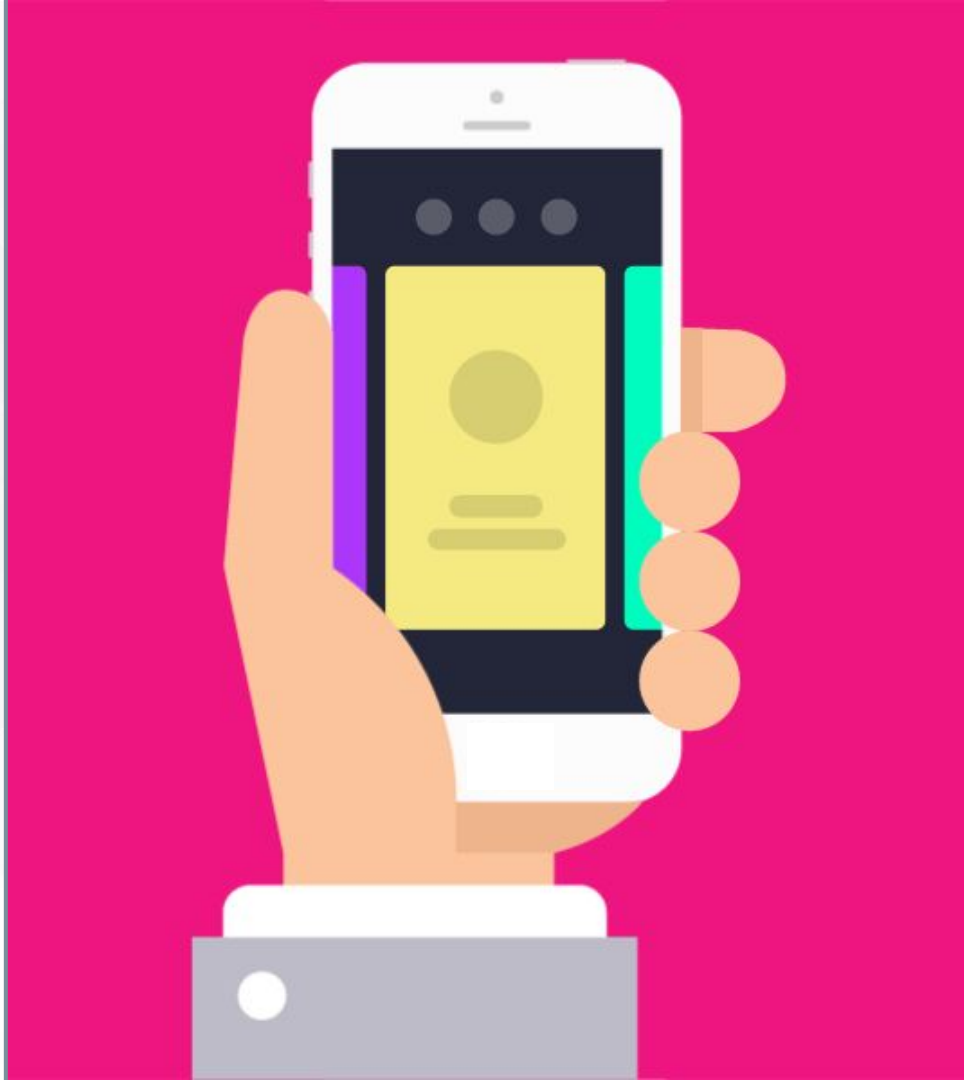
Fragments



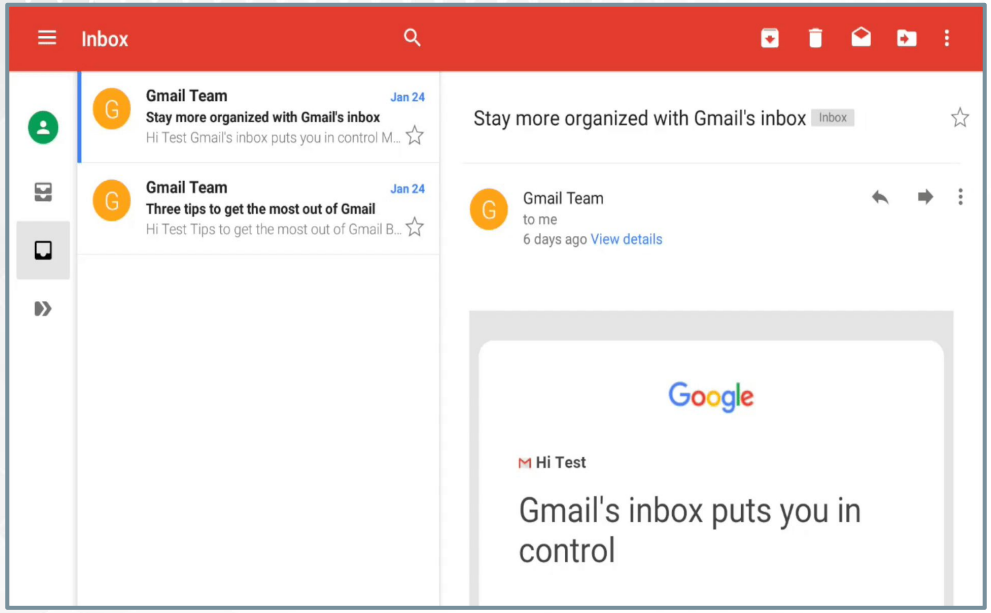
DigitalHouse > Coding School

O que são?

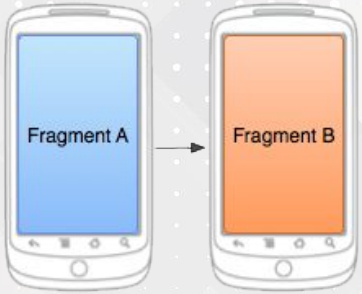
Um Fragment é um “pedaço” de tela que pode ser reutilizado em **outras partes** do seu app.

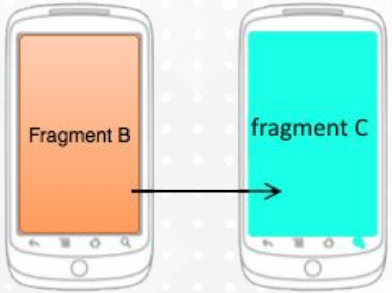


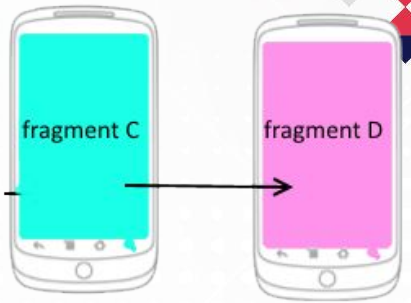
Como surgiu a necessidade?

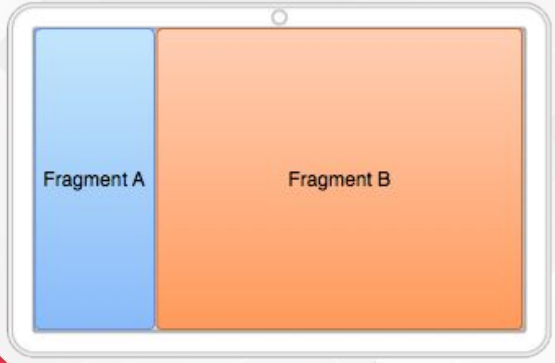


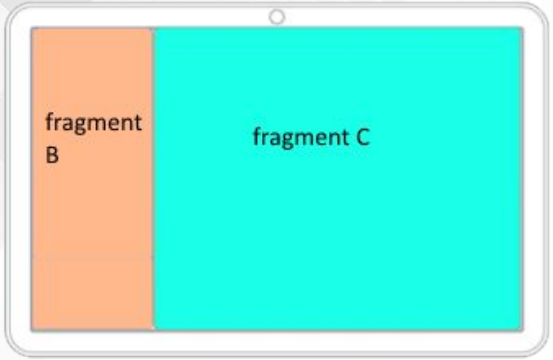
DigitalHouse > Coding School

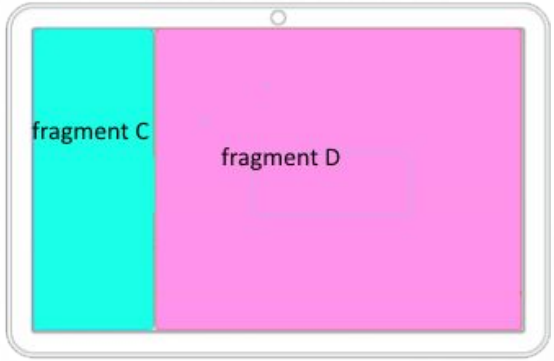












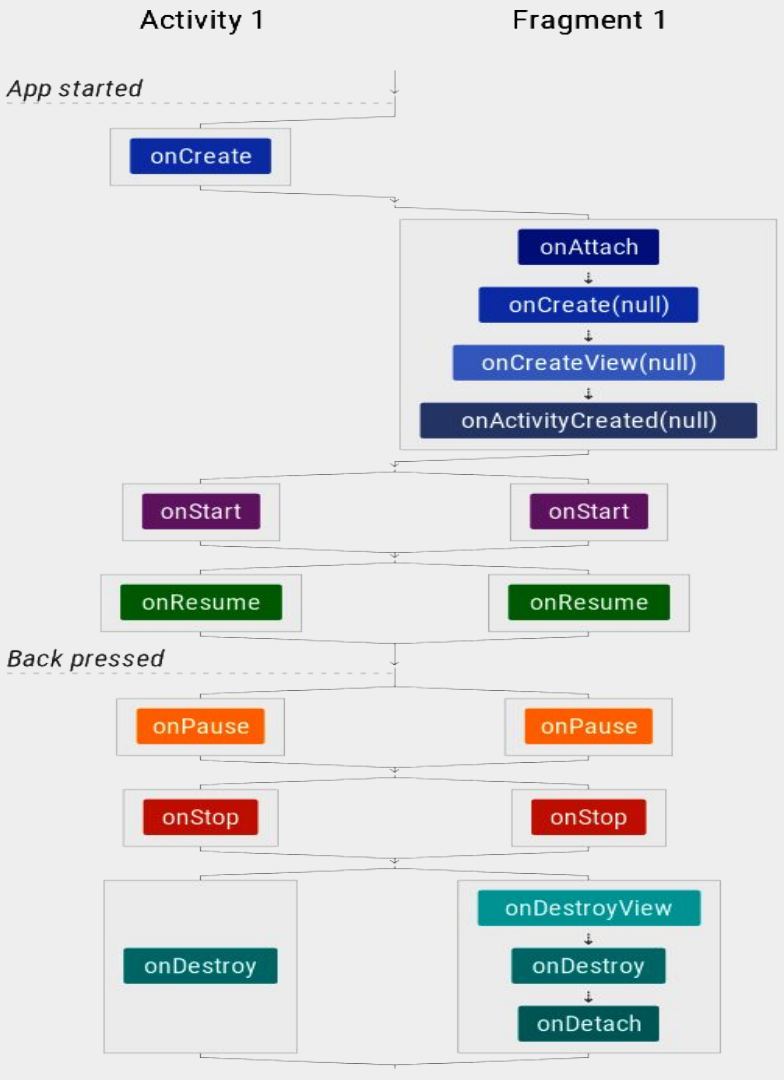
DigitalHouse > Coding School

Vantagens

- ➔ Reutilização de telas
- ➔ Diminui a duplicação de código
- ➔ Permite modularizar uma Activity
- ➔ Facilitação de designs dinâmicos e flexíveis para telas grandes

Ciclo de vida

- Ciclo de vida da **Fragment** depende do ciclo da **Activity**
- O ciclo começa quando a **Activity** está **ativa**
- É obrigatório a implementação do método **onCreateView**. Pois é invocado pelo sistema assim que a tela renderiza pela primeira vez



Ciclo de vida

onAttach - O onAttach é onde podemos obter uma referência para a Activity pai.

onCreate() - O sistema o chama ao criar o fragmento.

onCreateView() - O onCreateView é onde você constrói ou infla sua interface, faz conexão com alguma fonte de dados e retorna à Activity pai para poder integrá-lo em sua hierarquia de Views.

onActivityCreated() - Isso notifica nosso Fragment que a Activity pai completou seu ciclo no onCreate e é aqui que podemos interagir com segurança com a interface de usuário.

Ciclo de vida

onStart() - torna o fragmento visível para o usuário

onResume() - faz com que o fragmento comece a interagir com o usuário

onPause() - O sistema chama esse método como o primeiro indício de que o usuário está saindo do fragmento (embora não seja sempre uma indicação de que o fragmento está sendo destruído).

Ciclo de vida

onStop() - Assim como no ciclo de vida da Activity, uma vez que o Fragment não está mais visível, há uma chance dele ser encerrado.

onDestroyView() - O onDestroyView é correspondente ao onDestroy da Activity e é chamado imediatamente antes do Fragment ser destruido. Ele funciona independente da Activity pai.

onDestroy() - chamado para fazer a limpeza final do estado do fragmento.

onDeatch() - O onDetach é a última coisa que acontece no ciclo de vida, mesmo após o seu Fragment ser tecnicamente destruído.

Project
exemplo C:\Android\project\exemplo
.idea
app [exemplo-app]
libs
sampledata
src
androidTest
main
java
activity
base
binding
components
custom
dbroomkt
dbsqlite
debugging
drawer
facebook
firebase
fragment
functions
intent
layouts
mvvm
navdrawer
permissions
recycler
retrofit
revisao
share
telainicial

ActExemplo | Intent

```
class ActExemplo : AppCompatActivity() {

    //  Chamado na criação da activity
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        onViews()
        val extras : Bundle? = intent.extras

        val string : String? = extras?.getString( key: "chave0")
        val int : Int? = extras?.getInt( key: "chave1")
        val float : Float? = extras?.getFloat( key: "chave2")
        val double : Double? = extras?.getDouble( key: "chave3")
        val boolean : Boolean? = extras?.getBoolean( key: "chave4")
    }

    //  Após onStop, caso reiniciada.
    override fun onRestart() {
        super.onRestart()
    }

    //  Quando a activity se torna visível ao usuário
    override fun onStart() {
        super.onStart()
    }

    // Quando a activity fica disponível para interagir
    override fun onResume() {
        super.onResume()
    }

    // Quando a activity perde foco brevemente
    override fun onPause() {
        super.onPause()
    }

    // Quando a perda de foco leva a outra activity
```

DigitalHouse > Coding School

Diferenças	
Activity	**Fragment**
Só pode rodar uma por vez	Pode ter várias rodando ao mesmo tempo
Funciona por si só	Precisa de uma Activity para ser apresentado
Código único	Possibilidade de reutilização

Instanciando um Fragment numa Activity

Para colocarmos um **fragmento de tela** dentro de uma **Activity** precisamos iniciar uma **transação**, fazer o **replace** e **commitar**. Tudo isso através da classe **FragmentManager**:

```
val manager = supportFragmentManager
val transaction = manager.beginTransaction()
transaction.add(R.id.fragmentContainer, MeuFragment())
transaction.commit()
```

Exercício 1

Transições entre Fragments

Checklist:

- Criar um layout conforme o protótipo ao lado
- O cinza escuro representa a área onde o Fragment deve ser inserido
- Iniciar sem nenhum fragment
- Ao clicar no botão, abrir o Fragment referente a ele (terá dois fragments)
- Garantir que alternar entre eles irá funcionar sem quebrar o app

DigitalHouse >
Coding School