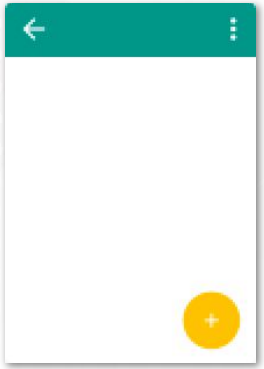
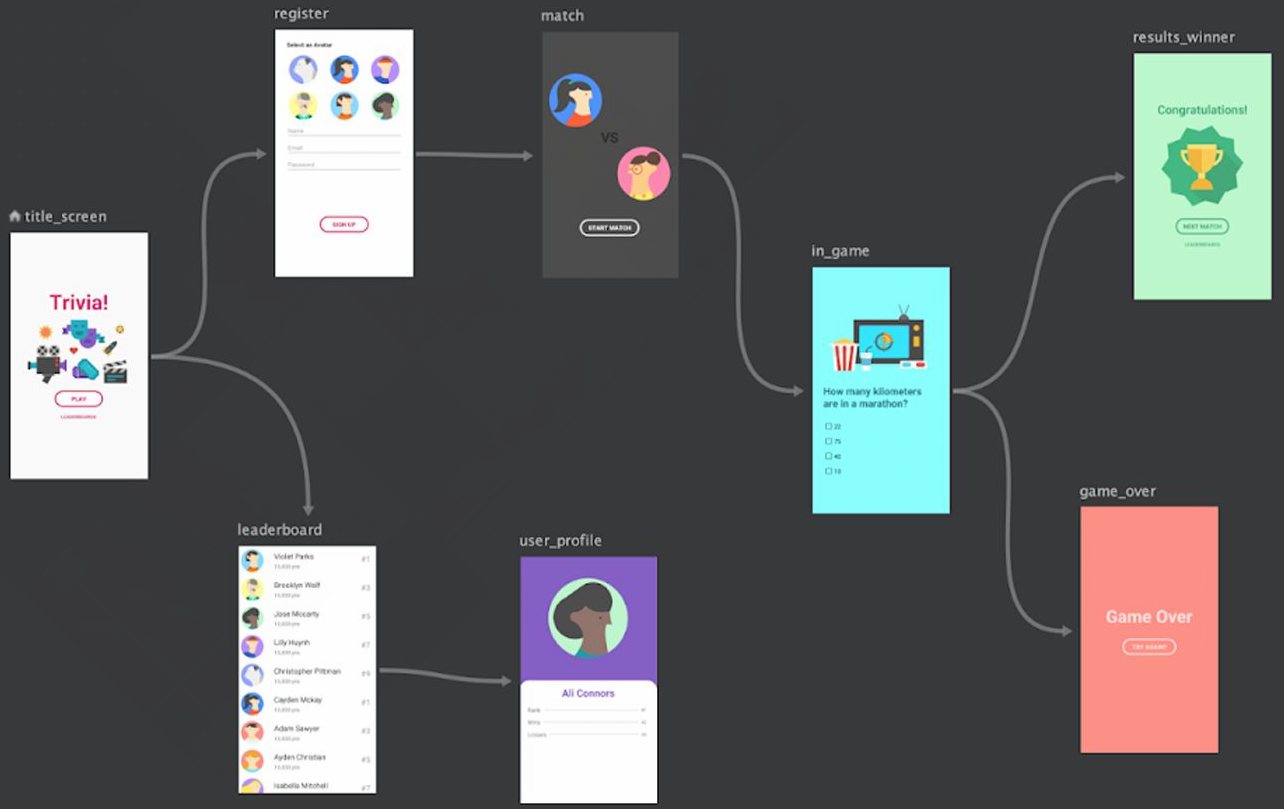
Aula 23 - Activity

DigitalHouse >
Coding School

Activity



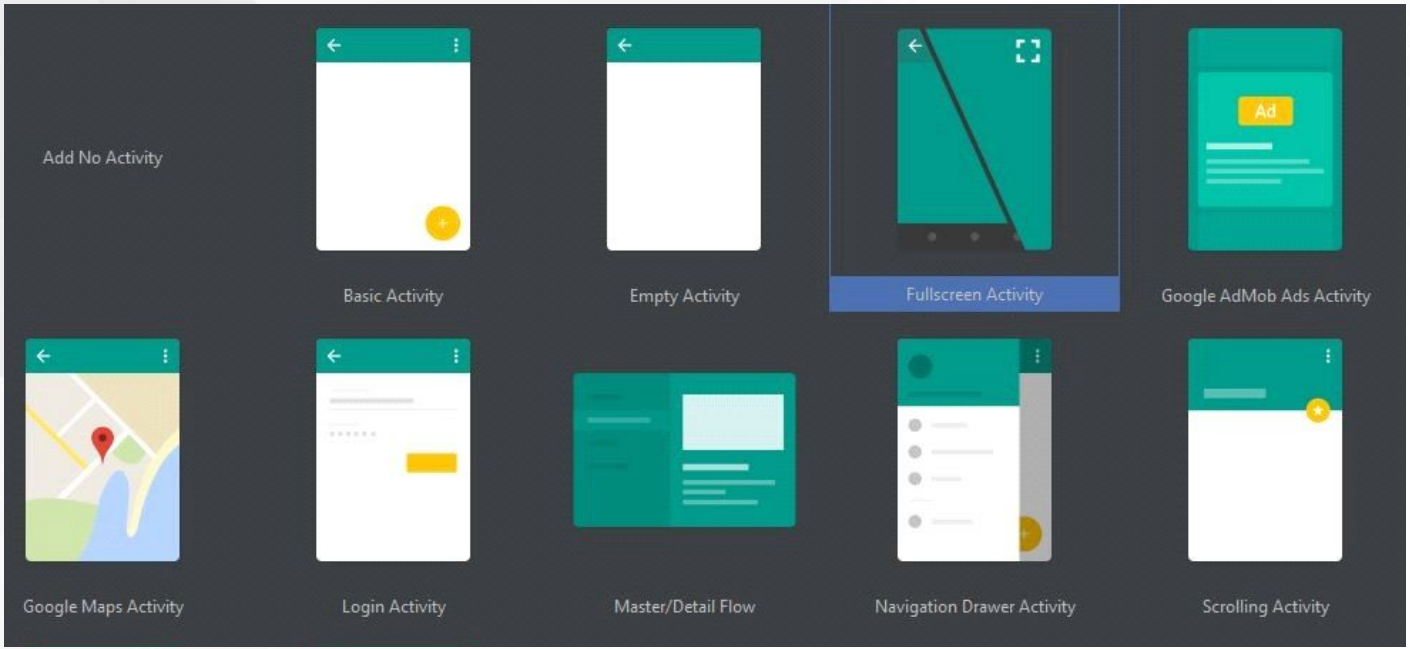

register
match
results_winner
title_screen
Trivia!
PLAY
SIGN UP
VS
START MATCH
Congratulations!
NEXT MATCH
in_game
How many kilometers are in a marathon?
leaderboard
Violet Parks
Brooklyn Wolf
Jose Mccarty
Lilly Huynh
Christopher Pittman
Cayden Mckay
Adam Sawyer
Ayden Christian
Isabella Mitchell
user_profile
Ali Connors
game_over
Game Over
TRY AGAIN!

As Activities são classes responsáveis pela apresentação visual de uma tela.

Cada Activity será uma tela diferente, cada uma começando como um quadro em branco pronto para receber conteúdo e interações.

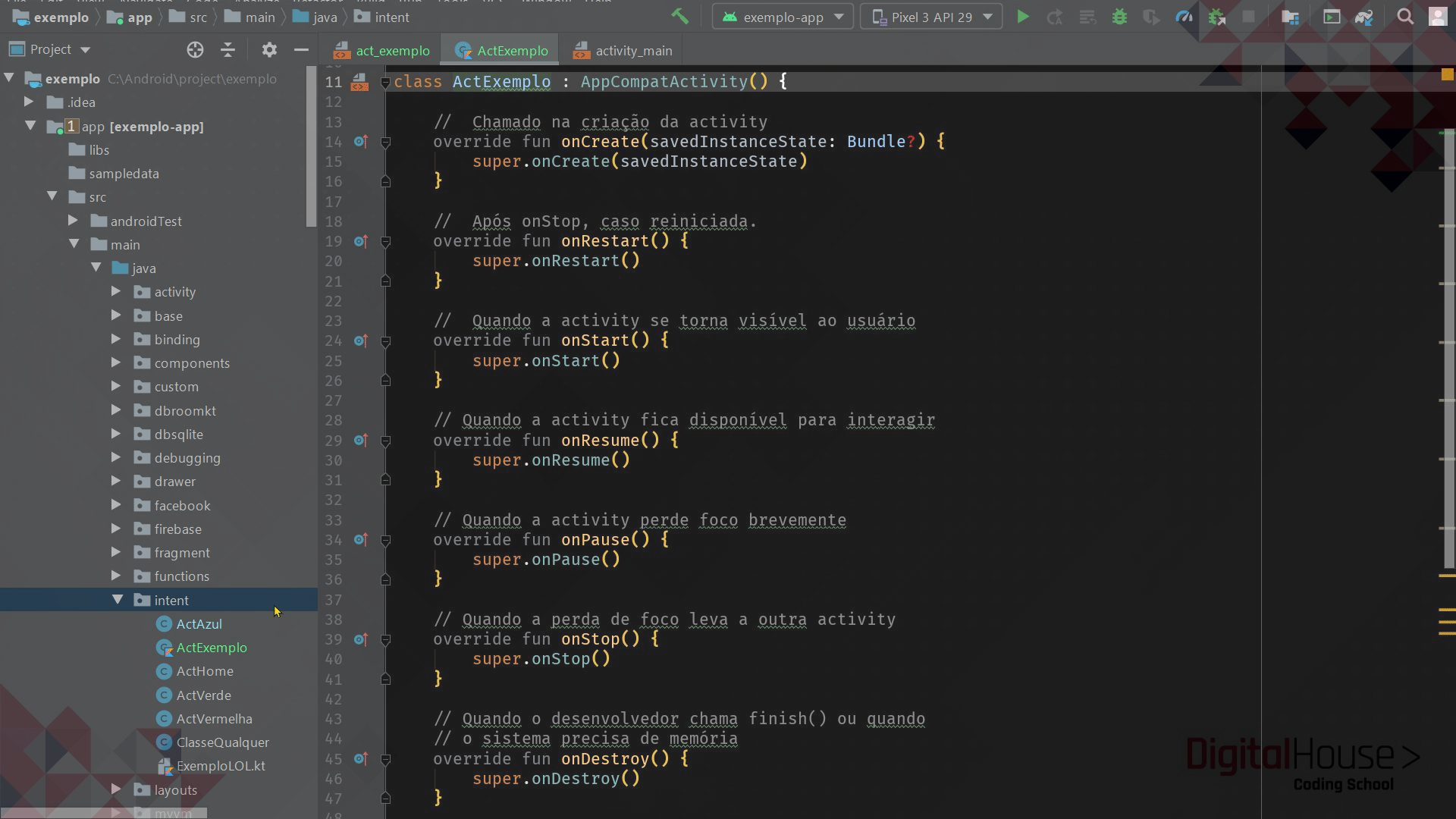
https://developer.android.com/reference/android/app/Activity

Criando uma Activity



https://developer.android.com/reference/android/app/Activity

DigitalHouse > Coding School


exemplo
app
src
main
java
intent
exemplo-app
Pixel 3 API 29
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
ActAzul
ActExemplo
ActHome
ActVerde
ActVermelha
ClasseQualquer
ExemploLOL.kt
layouts
act_exemplo
ActExemplo
activity_main
class ActExemplo : AppCompatActivity() {
// Chamado na criação da activity
override fun onCreate(savedInstanceState: Bundle?) {
super.onCreate(savedInstanceState)
}
// 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
override fun onStop() {
super.onStop()
}
// Quando o desenvolvedor chama finish() ou quando
// o sistema precisa de memória
override fun onDestroy() {
super.onDestroy()
}
DigitalHouse
Coding School

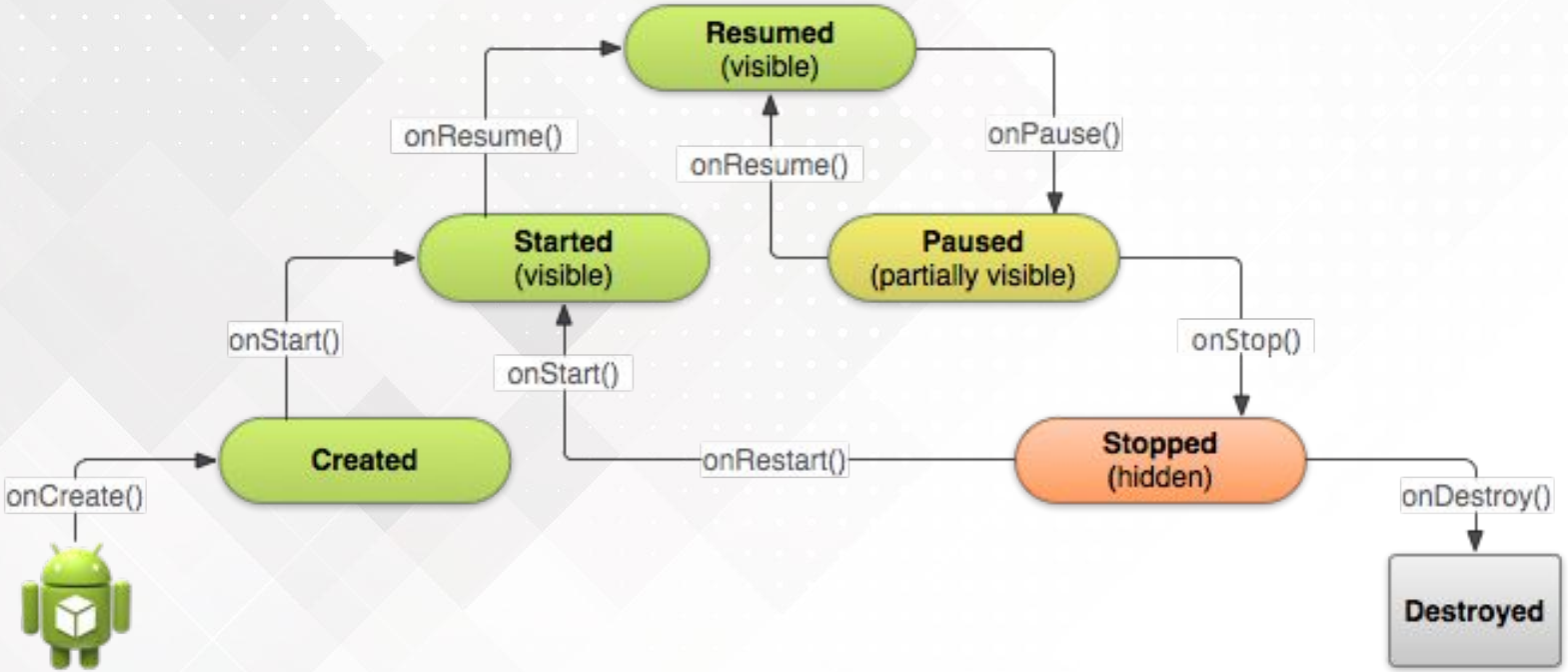
A classe base Activity da qual nossas Activities devem herdar, executa as funções ao lado toda vez que uma activity é criada e depois destruída.

Podemos sobrescrevê-las e adicionar comportamento no ato da criação, da retomada ou término de uma tela. Assim, estamos interagindo com o ciclo de vida da activity.

https://developer.android.com/reference/android/app/Activity

```kotlin
class ActExemplo : AppCompatActivity() {

    //  Chamado na criação da activity
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
    }

    //  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
    override fun onStop() {
        super.onStop()
    }

    // Quando o desenvolvedor chama finish() ou quando
    // o sistema precisa de memória
    override fun onDestroy() {
        super.onDestroy()
    }
```

Ciclo de Vida de uma Activity



https://developer.android.com/reference/android/app/Activity

Comunicando Dados entre Activities

Para iniciar outra tela/activity a partir da atual, precisamos declarar um novo objeto de intenção. Esse objeto recebe em seu construtor dois argumentos: de qual activity essa intenção vai partir e para qual ela vai.

Ao chamar a função startActivity informando tal Intent, a Activity será iniciada, conforme no click do botão demonstrado abaixo.

```
botao.setOnClickListener { it: View!
    println("Olá, Android!")
    val intent = Intent( packageContext: this, ActHome::class.java)
    startActivity(intent)
}
```

Também é possível inserir dados extras na intent tal como em um mapa. Tais serão enviados à activity que está sendo criada.

```
botao.setOnClickListener { it: View!
    println("Olá, Android!")
    val intent = Intent( packageContext: this, ActHome::class.java)

    intent.putExtra( name: "chave0", value: "valor")
    intent.putExtra( name: "chave1", value: 0)
    intent.putExtra( name: "chave2", value: 0f)
    intent.putExtra( name: "chave3", value: 0.0)
    intent.putExtra( name: "chave4", value: false)
    //etc

    startActivity(intent)
}
```

Os dados podem ser recuperados de dentro da Intent na activity que foi iniciada, tal como abaixo. Cuidado com a nulabilidade!

```
//  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")
}
```


The Business:
THE APP ISN'T WORKING!
imgflip.com
Developer:
WORKS FINE ON MY MACHINE!