

FERRAMENTAS E AMBIENTES DE DESENVOLVIMENTO

Conhecendo e explorando as principais ferramentas utilizadas na criação de Apps.

INTRODUÇÃO

Ambientes de Desenvolvimento;
Aplicações, Ferramentas e Utilitários.



BREVE HISTÓRIA DOS AMBIENTES DE DESENVOLVIMENTO

Como era nos primórdios?



Terminais de texto;
Editores simples;
Ferramentas de linha de comando;
Grande tempo de compilação.



O que são IDE's?

Integrated Development Environments

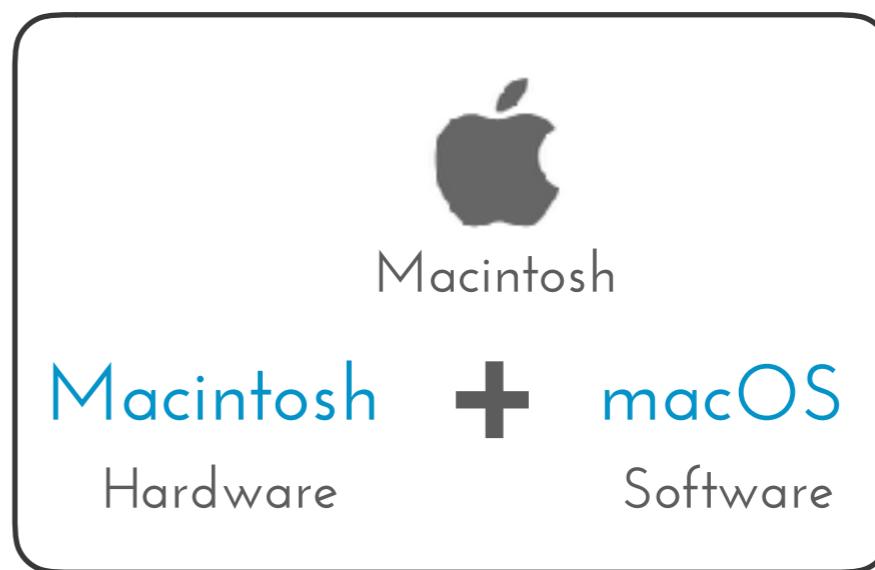
Sigla de Integrated Development Environments (Ambiente de Desenvolvimento Integrado)

Uma ferramenta que reune os principais recursos necessários para Desenho (Design), Programação (Programming), e Distribuição (Deployment) de programas para uma determinada plataforma.

O que são Plataformas?

Um conjunto de tecnologias na qual podemos criar programas.

Geralmente uma composição de hardware e software, exemplos:



O que são Plataformas?

Mas pode se referir a uma abstração do conceito de hardware, exemplos:

Plataforma Web: grupo de tecnologias usado para criar programas que rodam em Web Browsers;

Java/.NET: plataforma para criação de aplicações corporativas.

Android: plataforma para criação de Apps que executam em dispositivos móveis;

Xamarin: plataforma para criação de Apps em diversas



O que é uma API?

Application Programming Interface

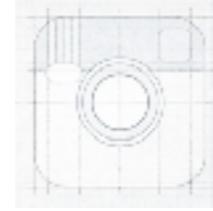
Sigla de Application Programming Interface (Interface de Programação da Aplicação)

Toda interface ou biblioteca que podemos aproveitar em nossos programas para usar recursos de uma outra aplicação.



O que é uma API?

Application Programming Interface



Exemplos:

[Facebook API](#) - Interagir com o Facebook

[Instagram API](#) - Interagir com o Instagram

[Google Maps API](#) - Integrar mapas em suas soluções

[IBM Watson](#)

O que é um SDK?

Software Development Kit

Sigla de Software Development Kit (Kit de Desenvolvimento de Software).

Conjunto (kit) de ferramentas fornecidas pelo desenvolvedor de uma plataforma que permitem a criação de programas para serem executados nela,



O que é um SDK?

Software
Development
Kit

Exemplos:

Windows SDK: utilizado para criação de programas nativos para Windows.

iOS SDK: utilizado para criação de Apps nativos para a plataforma iOS.

Android SDK: utilizado para criação de Apps nativos para a plataforma Android.



Principais IDE's do Mercado

Android Studio (Android)



Eclipse (Java, C, C\++, etc)



IntelliJ IDE (Java, C, C\++, etc)



NetBeans (Java, C, C\++, etc)



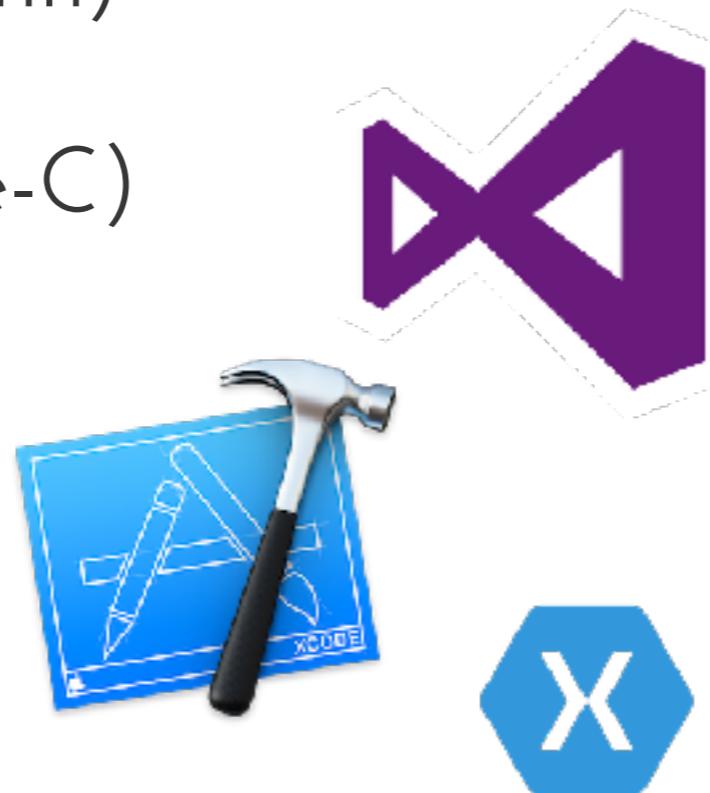
Principais IDE's do Mercado

Visual Studio (.NET, C#, Visual Basic, C, C\++, etc)

Xamarin Studio (Xamarin)

Xcode (Swift, Objective-C)

Zend Studio (PHP)



Preciso de IDE's?

?

?

Não, mas...

?

?

APLICAÇÕES, FERRAMENTAS E UTILITÁRIOS

Programas vs. Aplicativos vs. Aplicações vs. Apps

Programas

Executam uma operação com uma entrada e saída previsíveis. São a unidade básica da computação.



Programas vs. Aplicativos vs. Aplicações vs. Apps

Aplicativos

Programas complexos que rodam em algum tipo de plataforma e que oferecem soluções para um tipo específico de problemas.



Exemplos

Microsoft Office: Word, Excel, PowerPoint, etc.

Programas vs. Aplicativos vs. Aplicações vs. Apps

Aplicações

Um conjunto de aplicativos que geralmente envolve diversas partes ou plataformas com o objetivo de fornecer uma solução para um problema ou rotina complexos.

Aplicações de CRM (Salesforce, Dynamics, etc.)

Aplicações de ERP (SAP, Microsoft AX, etc.)

Aplicações de Gestão de Fluxo de Caixa



Programas vs. Aplicativos vs. Aplicações vs. Apps

Apps

Uma abreviação de **Applications** que tornou-se popular com a introdução do iOS e da App Store.



CONHECENDO AS APLICAÇÕES

Invision

Zeplin

HockeyApp



InVision



Plataforma para criação de
Protótipos Interaivos de Apps.

Permite compartilhamento
fácil com a equipe e os
equipes.

Inclui ferramenta de
comentários para facilitar
discussões e ajustes.



InVision



Inclui ferramenta para apresentação em tempo real.

Tem Apps que podem ser usados para abrir os protótipos e simular seu funcionamento.



Demo

ZEPLIN



Plataforma para
compartilhamento dos
Wireframes.

Permite visualizar facilmente
todas as características técnicas
de uma tela.

Permite a exportação fácil dos
assets de uma tela.

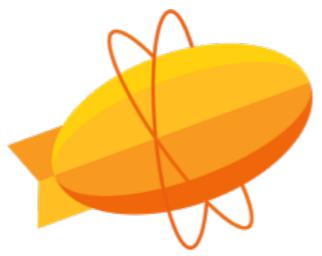


ZEPLIN



Permite a criação de Guias de Estilos que facilitam a criação de temas dos Apps

Inclui ferramenta de comentários para facilitar discussões e ajustes.



Demo

Hockeyapp



Plataforma para distribuição de versões de testes de Apps.

Permite criar times de distribuição com funções pré-determinadas.

Inclui recursos para coleta e visualização de relatórios de erros.

Hockeyapp



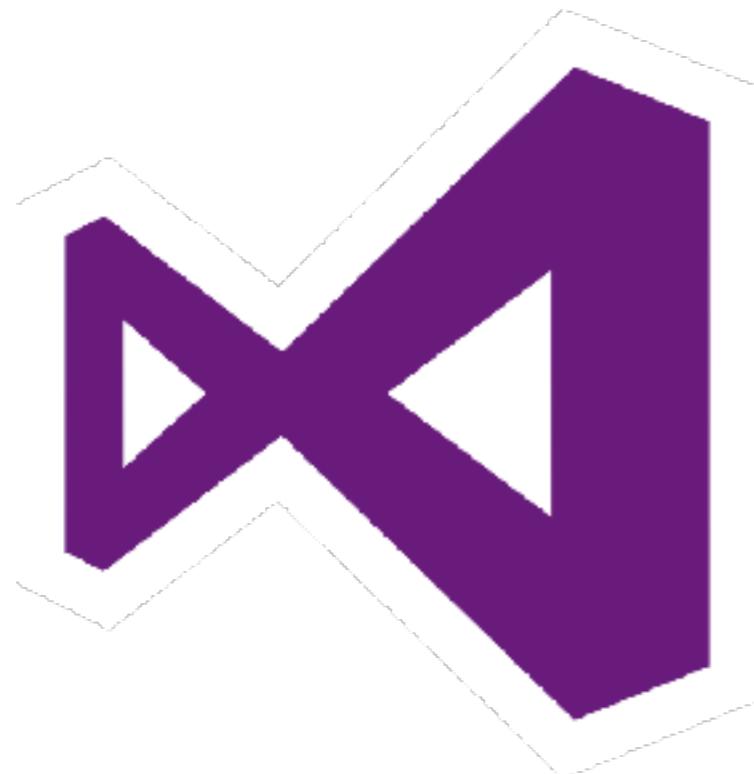
Inclui uma ferramenta integrada de feedback que permite aos testadores se comunicarem com o time.

Inclui diversos relatórios que permitem visualizar eventos relacionados à distribuição.



Demo

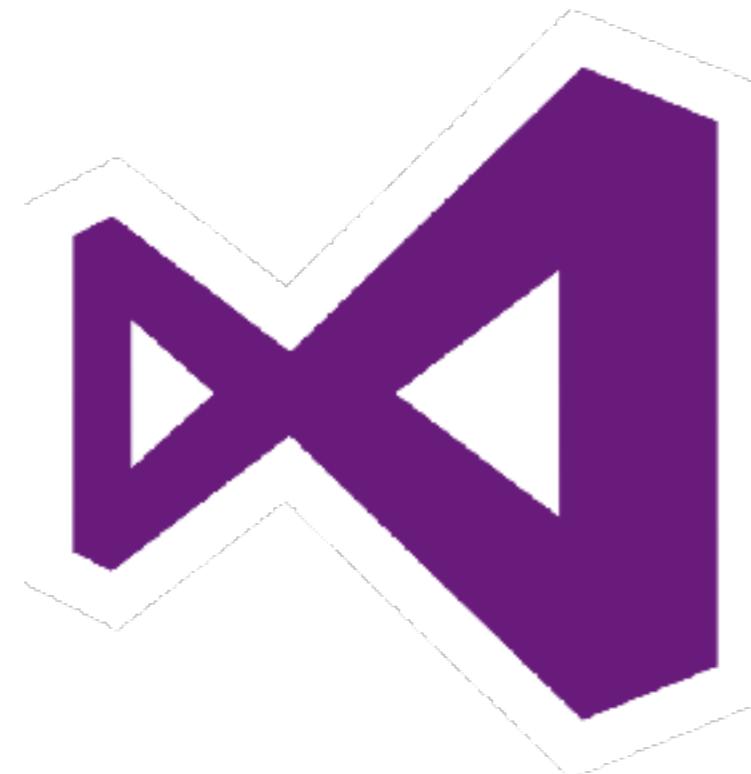
Visual studio team services



Plataforma de Gerenciamento
de Projetos de Software.

Inspirado no modelo Agile.

Visual studio team services



Inclui ferramentas para todo o ciclo de desenvolvimento:

Gerenciamento de Requisitos

Gerenciamento de Código Fonte

Gerenciamento de Builds e Releases

Gerenciamento de Testes



Demo

CONHECENDO FERRAMENTAS E UTILITÁRIOS

Sketch

PaintCode

Postman

Reflector

ColorSnapper

SourceTree

GitKraken



Sketch



Ferramenta para criação de
Protótipos de Apps.

Exclusivo para macOS.

[Link](#)

Postman



Utilitário para testar a comunicação com serviços de Internet.

Será mais explorado no módulo sobre Apps Conectados.

Multiplataforma (Windows, macOS).

[Link](#)

Reflector



TM

Utilitário para espelhar a tela
do celular no computador.

[Link](#)

ColorSnapper



Utilitário para obter a cor de qualquer parte da tela.

[Link](#)

SourceTree



Ferramenta bastante completa para gerenciamento de repositórios GIT.

Multiplataforma (Windows, macOS).

[Link](#)

GitKraken



Ferramenta moderna para gerenciamento de repositórios GIT.

Multiplataforma (Windows, macOS).

[Link](#)

Pixelmator



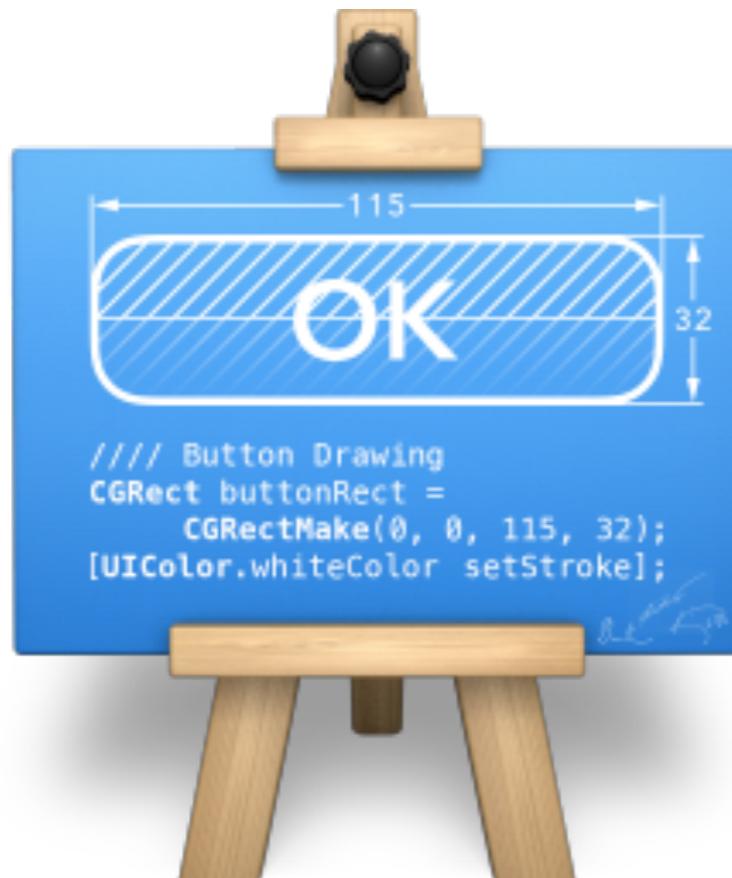
Um editor gráfico com recursos próximos ao do Photoshop.

Mais barato, sendo um bom substituto.

Exclusivo para macOS.

[Link](#)

PaintCode



Aplicativo de desenho que transforma a imagem em código que a representa.

Ajuda a otimizar os recursos dos Apps.

Exclusivo para macOS.

[Link](#)

OUTROS RECURSOS

Microsoft Windows Azure

Visual Studio Developer Essentials



AZURE

Microsoft Azure

Plataforma de Computação
em Nuvem (Cloud
Computing) da Microsoft

Oferece dezenas de tipos de
serviços comuns a Aplicações.

Será explorado no módulo
sobre Apps Conectados.



AZURE

Microsoft Azure

Oferece dezenas de recursos
gratuitos para desenvolvedores

Link



Visual Studio Developer Essenciais



Oferece dezenas de recursos gratuitos para desenvolvedores.

[Link](#)

CONHECENDO AS IDE'S

Android Studio

Xcode



Android Studio



Apresentando o Android
Studio.

Origem e Breve História.

Java e Android.

Android Studio



Entendendo o JDK.

Entendendo o ADT.

Emuladores de Dispositivos.

[Link](#)



Xcode



Apresentado o Xcode;
Origem e Breve História;
NextStep e Apple;
Linguagens de Programação:
Objective-C
Swift

[Link](#)

PREPARANDO O AMBIENTE

EXPLORANDO AS FERRAMENTAS

APRESENTANDO O ANDROID STUDIO

O que é o Android Studio?

Android Developer Tools (ADT)

Instalando o Android Studio

Principais recursos



O que é o Android Studio?

IDE oficial do Google para desenvolvimento Android.

Personalização do IntelliJ IDE.

Inclui o ADT e diversas ferramentas próprias para desenvolvimento Android

Substitui o Eclipse com ADT.



Android Developer Tools (ADT)



Um pacote contendo os compiladores e outras ferramentas para geração de programas Android.

O ADT é independente do Android Studio.

O Android Studio depende do ADT e o instala automaticamente.

Instalando o Android Studio

Pré-Requisitos:

Disponível para Windows, OSX e Linux;

JDK 1.7 ou superior;

Verifique o guia de instalação do Android da Caeno

[Link](#)

Principais Recursos

Editor de código avançado;

Templates de Projeto;

Templates de Código e Recursos;

Integração com VCS (Git, GitHub, SVN, etc);

Editor de Layouts;

Ferramentas de Debug (ADB);

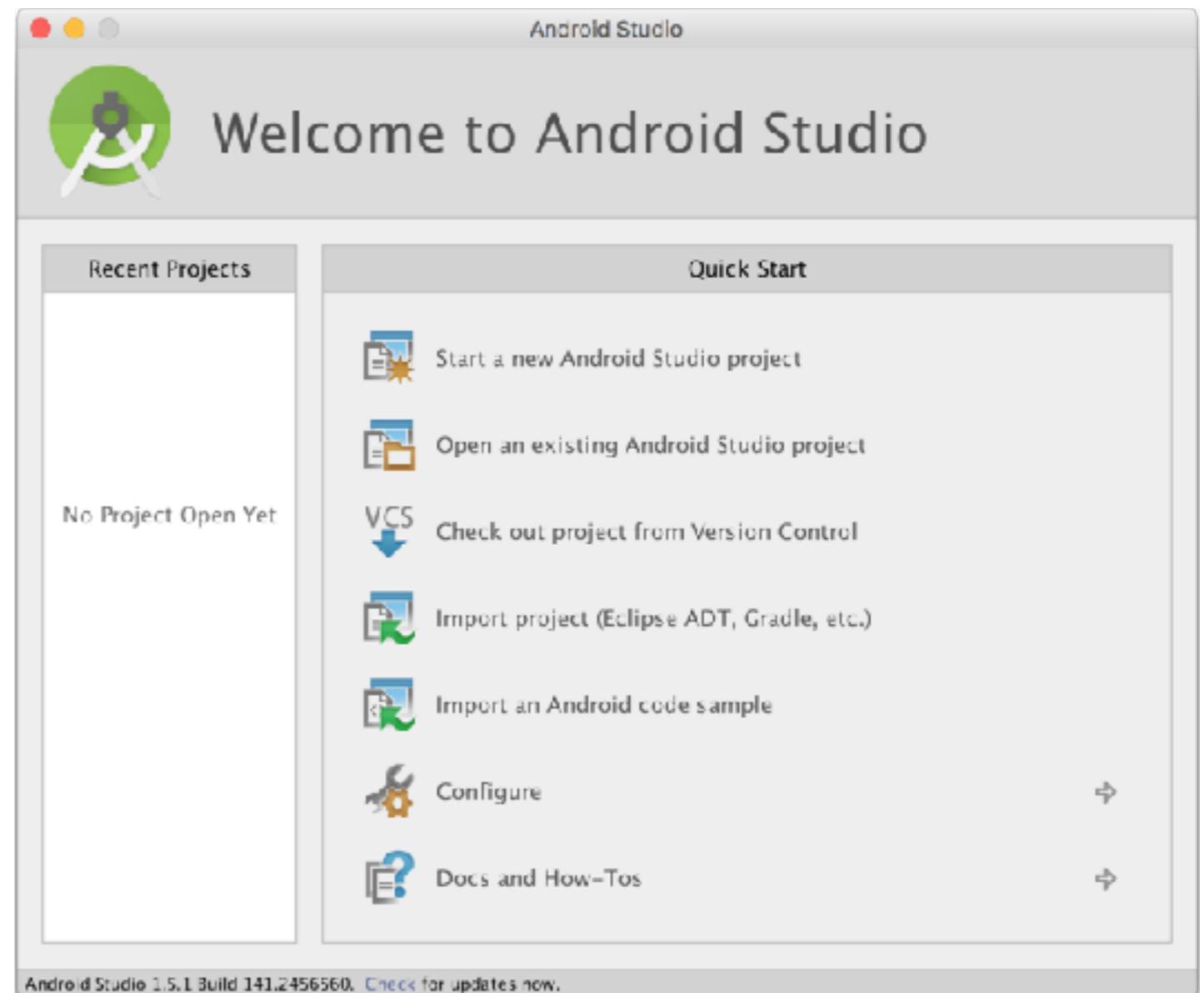
Emuladores Android.

CONHECENDO O ANDROID STUDIO

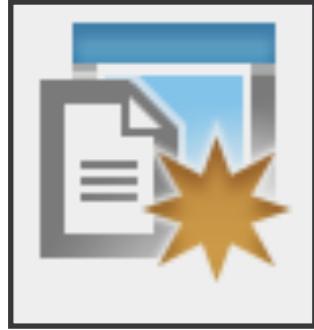
Tela de Boas-Vindas
Estrutura da Tela Principal
Estrutura de um Projeto Android
Demo
Exercício

Tela de Boas-Vindas

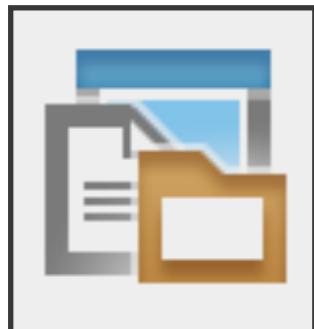
O Android Studio abre com a tela de boas-vindas que dá acesso aos seus principais recursos.



Resumo das Opções



Start a New Android Studio Project: cria um novo projeto Android usando um dos templates disponíveis.



Open an existing Android Studio Project: abrir um projeto existente.

Resumo das Opções



Check out project from Version Control:
permite abrir um projeto a partir de um repositório de controle de código (VCS) usando uma das ferramentas suportadas: Git/GitHub, Mercurial, SVN (Subversion), etc.

Resumo das Opções



Import project (Eclise ADT, Gradle, etc.):
permite abrir ou importar um projeto existente. Essa opção é preferível à segunda, já que ela identifica automaticamente uma pasta de projeto Android.

Resumo das Opções



Import an Android Code sample: há uma infinidade de códigos de exemplo publicados no site oficial do Android que podem ser importados automaticamente através dessa ferramenta.

Resumo das Opções



Configure: permite acessar diversas opções do ambiente e da ferramenta (será explorado posteriormente).



Docs and How-Tos: permite acessar as documentações do Android Studio e da plataforma Android.

Estrutura da Tela Principal

Barra de Ferramentas (Toolbar)

Editor

Barra de Navegação (Navigation Bar)

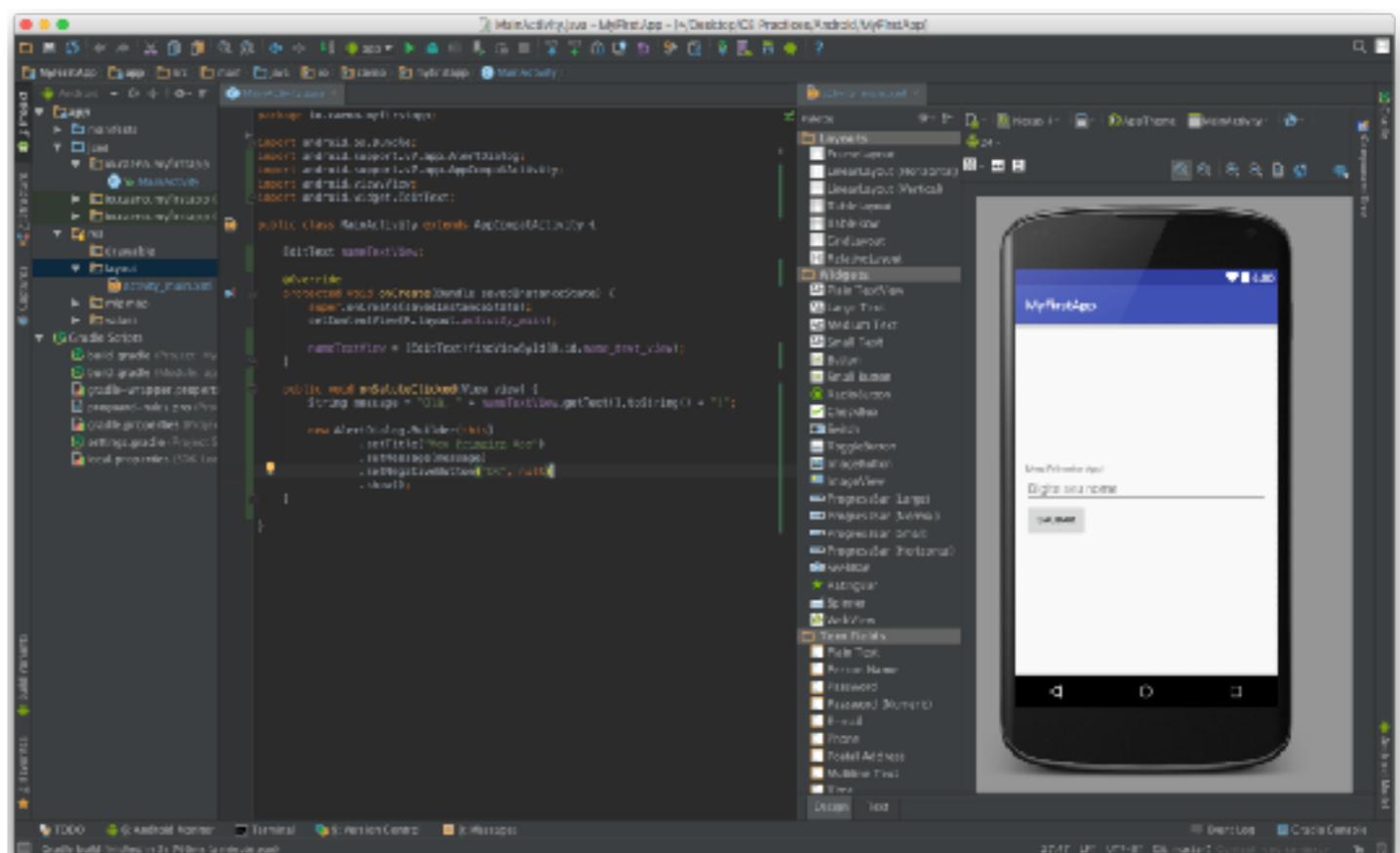
Tools Windows

Project

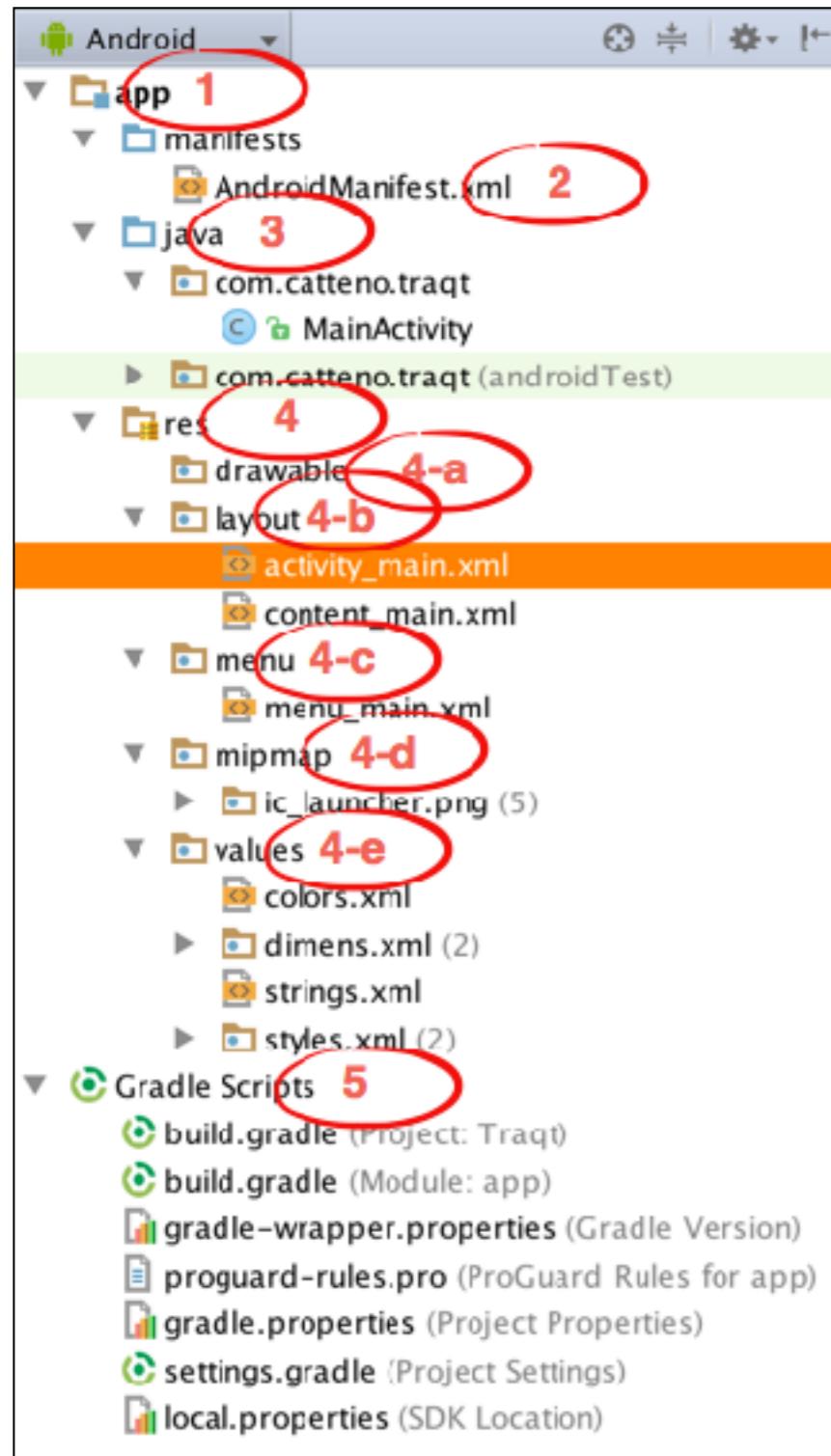
Structure

Debug

Event Log



Estrutura de um Projeto Android



1. Módulos (app)

fontes e resources do App

2. Manifests

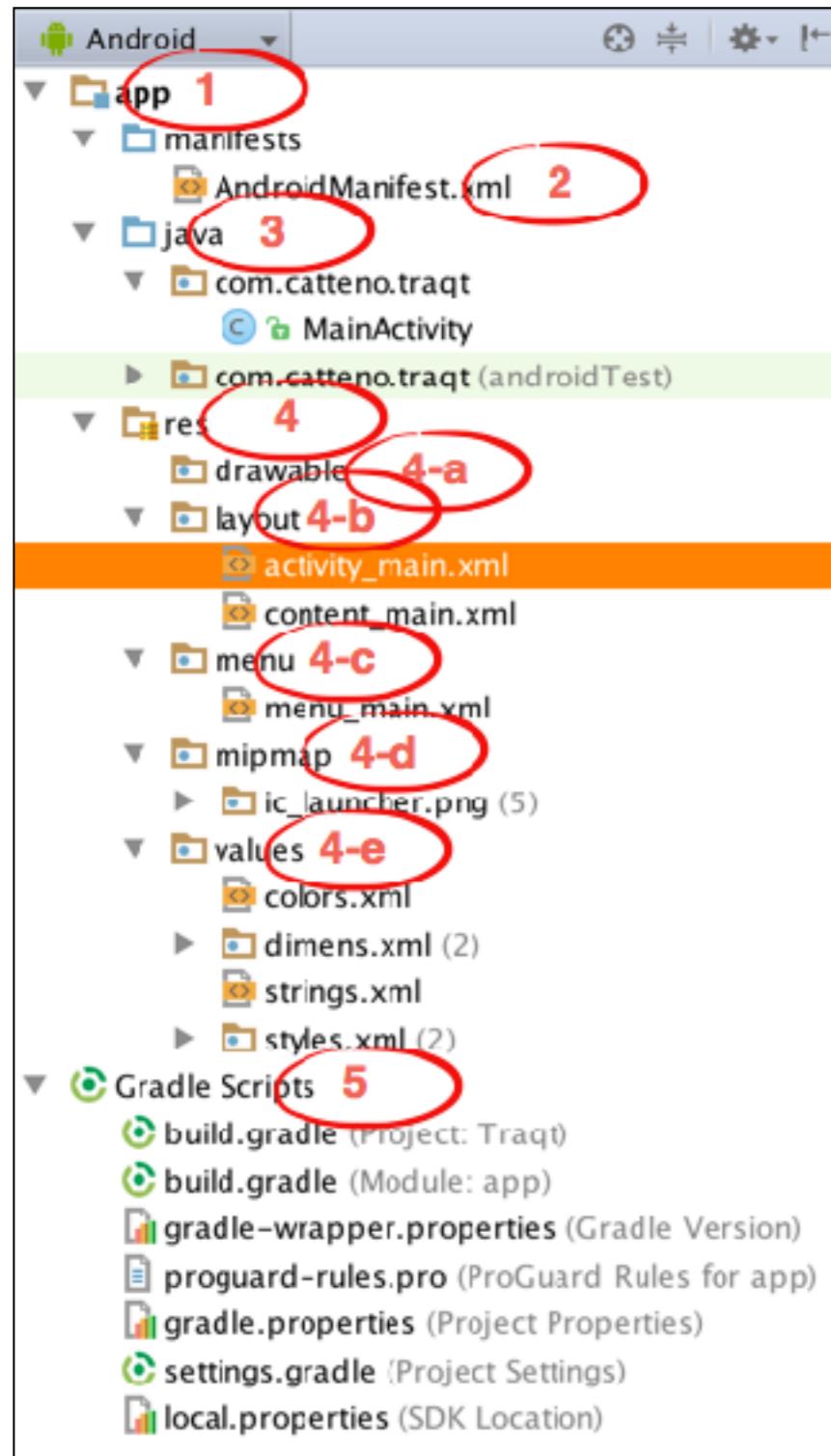
contém o **AndroidManifest.xml**

controla configurações gerais do App

3. java

código fonte

Estrutura de um Projeto Android



4. res

a. drawable

imagens ou vetores

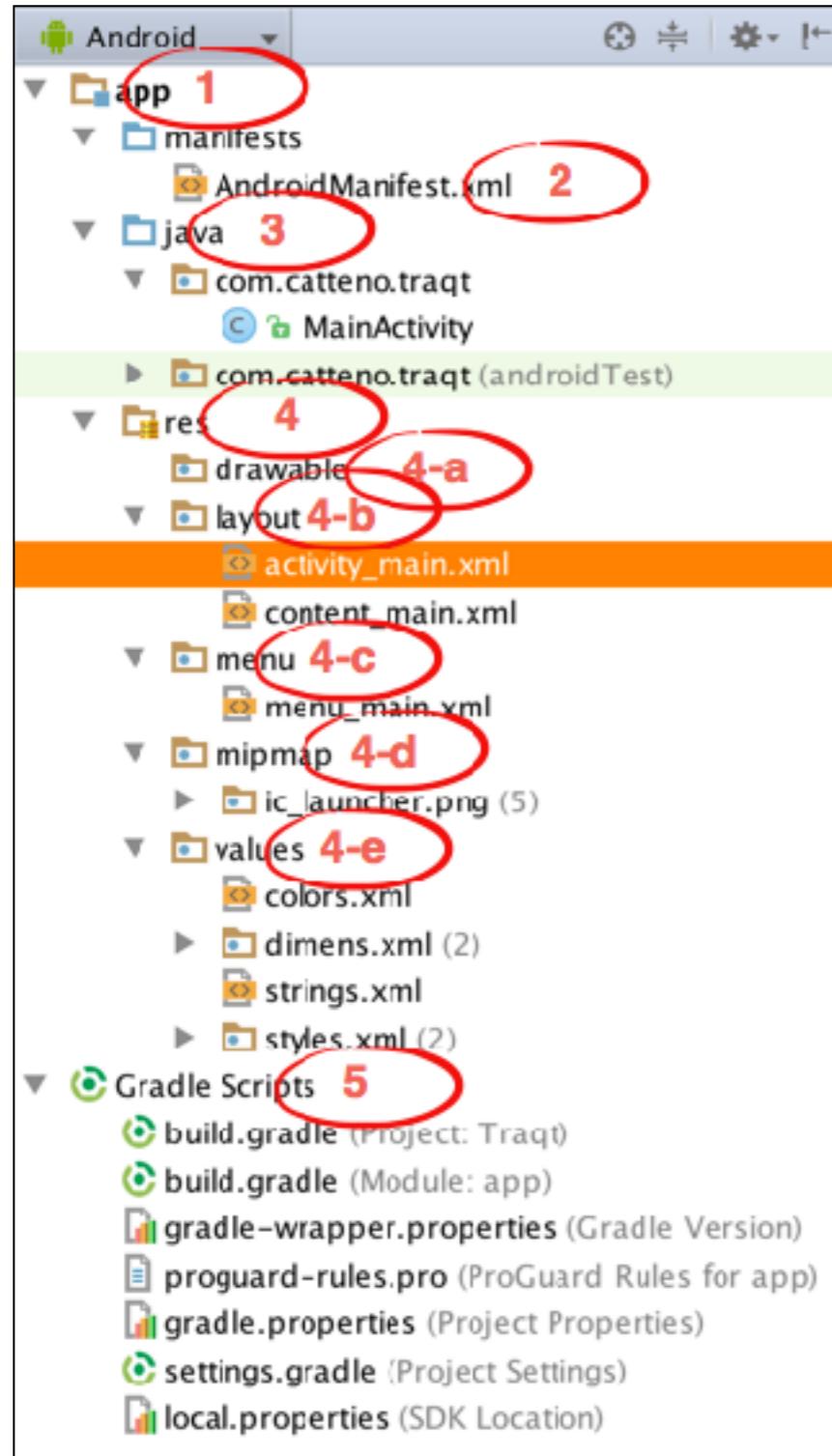
b. layout

recursos que descrevem interfaces

c. menu

recursos que descrevem

Estrutura de um Projeto Android



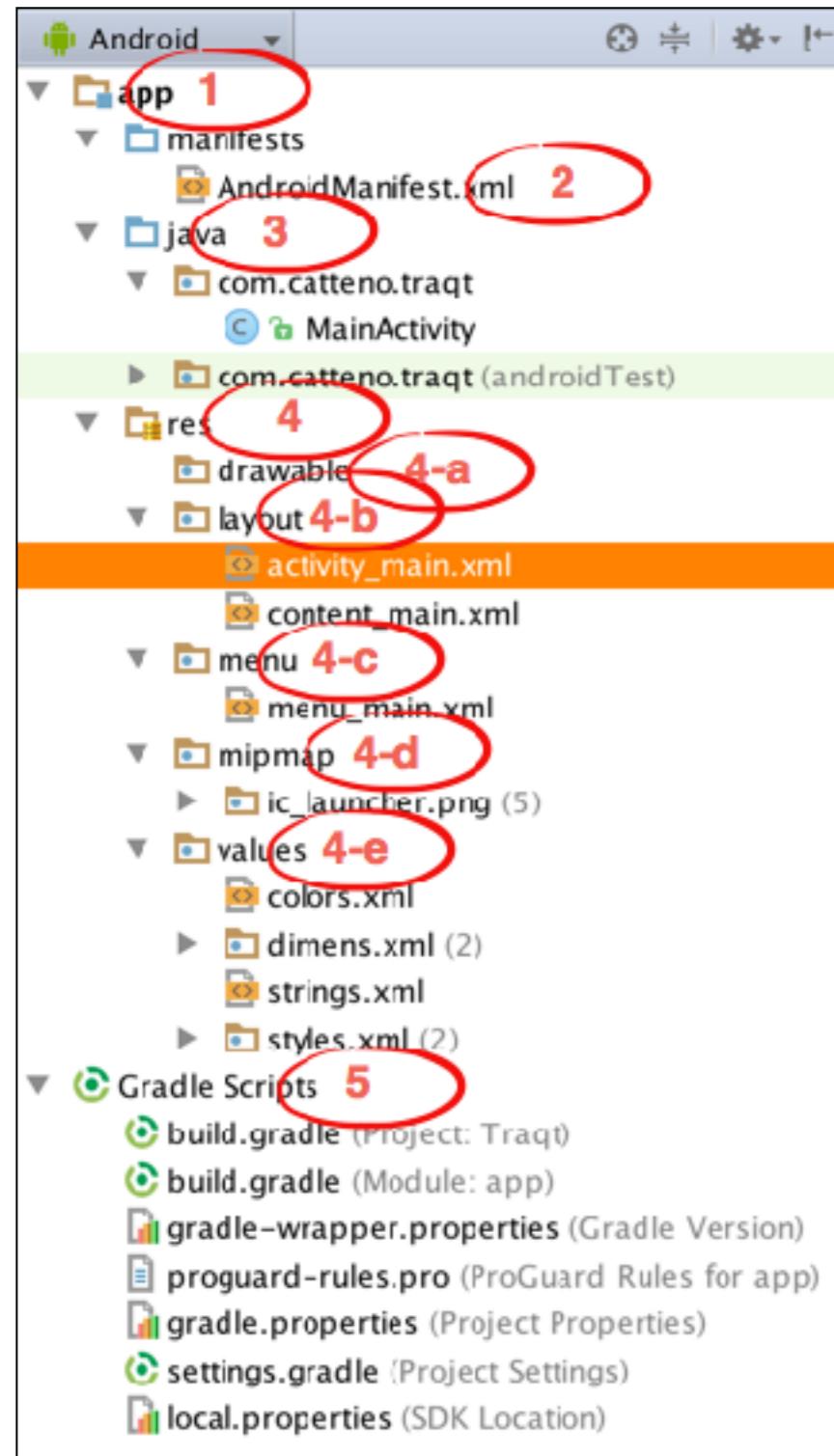
4. res

d. mipmap:

Ícones do Launcher

e. values

Estrutura de um Projeto Android



outros tipos de recursos:

strings

cores

dimensões

3. CONFIGURANDO O ANDROID STUDIO

SDK Platforms

SDK Tools

Usando o SDK Manager

SDK Platforms

Entendendo os API Levels

Cada release do Android é identificado por:

- a) Seu código de versão.
- b) API Level.

SDK Platforms

a) Código de Versão

Geralmente no formato X.Y.Z onde:

major

Mudanças significativas na plataforma

Introduz novas API's

Geralmente quebram a
compatibilidade com as anteriores

minor:

Inclui novos recursos e API's sem
quebrar compatibilidade.

revision:

Correções de bugs

X - major

Y - minor

Z - revision

Exemplos:

2.2

2.4.2

5.0.1



SDK Platforms

b) API Level

Um número inteiro sequencial

O API Level nos ajuda a identificar quais bibliotecas estão disponíveis para desenvolvimento.

Todo projeto deve configurar qual a versão mínima necessária.

De acordo com a versão mínima o Android Studio sinaliza a disponibilidade de recursos.

SDK Tools

Ferramentas e compiladores do Android.

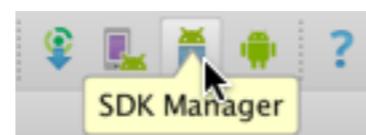
Bibliotecas adicionais.

Usando o SDK Manager

O SDK Manager gerencia a instalação dos SDK Platforms e SDK Tools.

Selecionamos os componentes de acordo com as nossas necessidades de desenvolvimento.

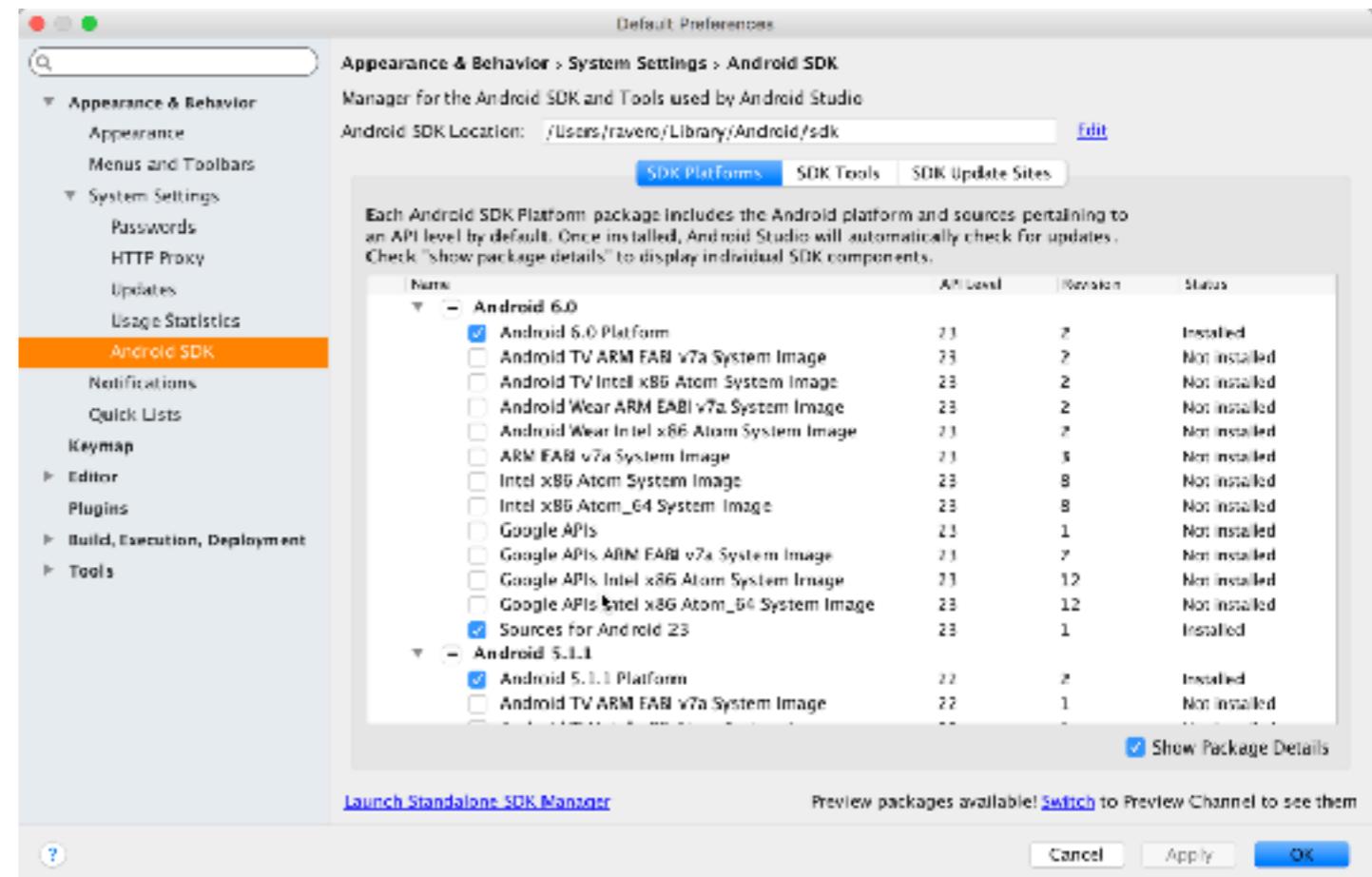
Acessado através da Toolbar no segmento final.



Usando o SDK Manager

SDK Platforms

Componentes dos API Levels do Android.

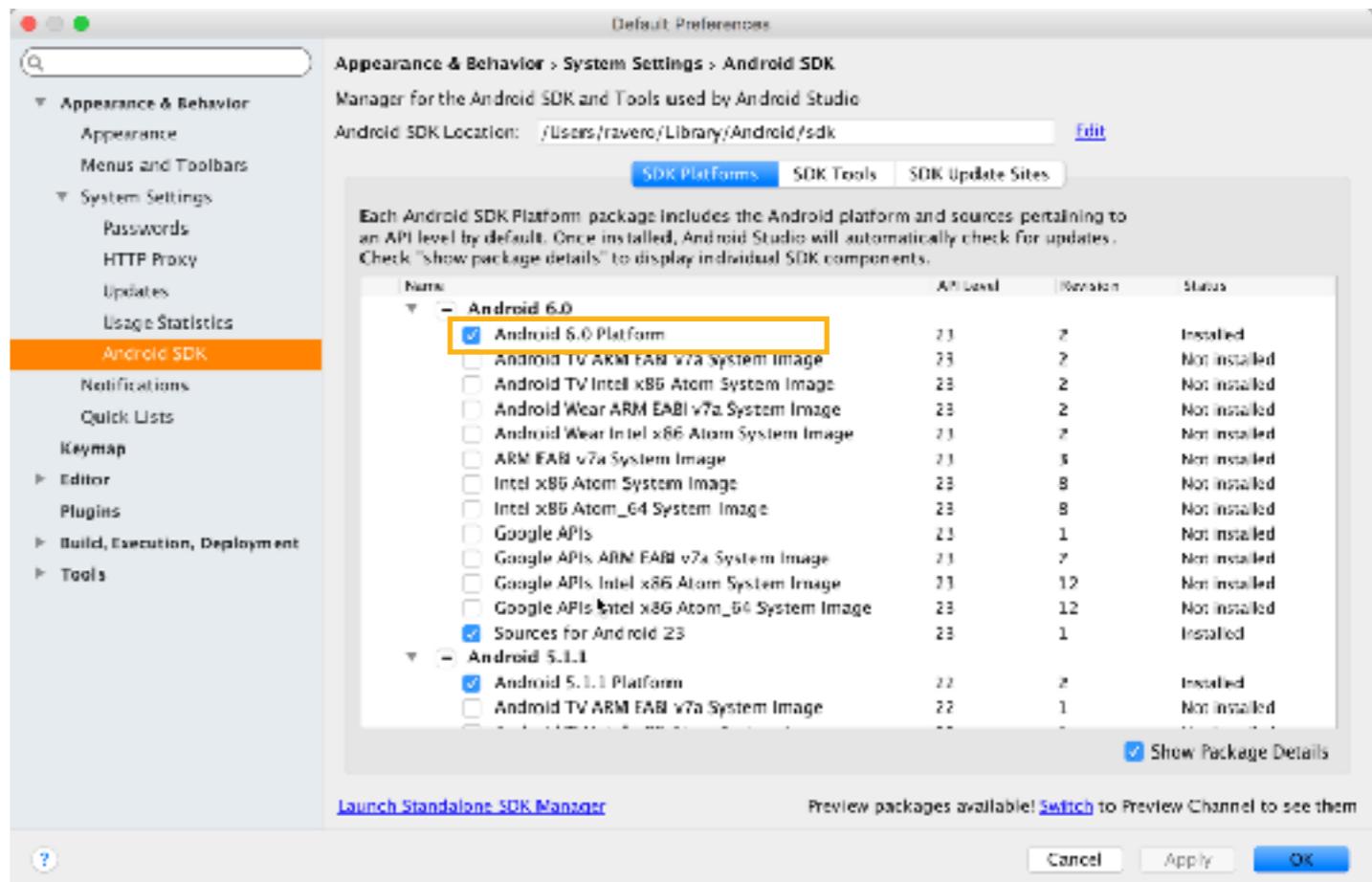


Usando o SDK Manager

Principais Componentes:

Android Platform

O componente básico para incluir suporte a um API Level no Android Studio.



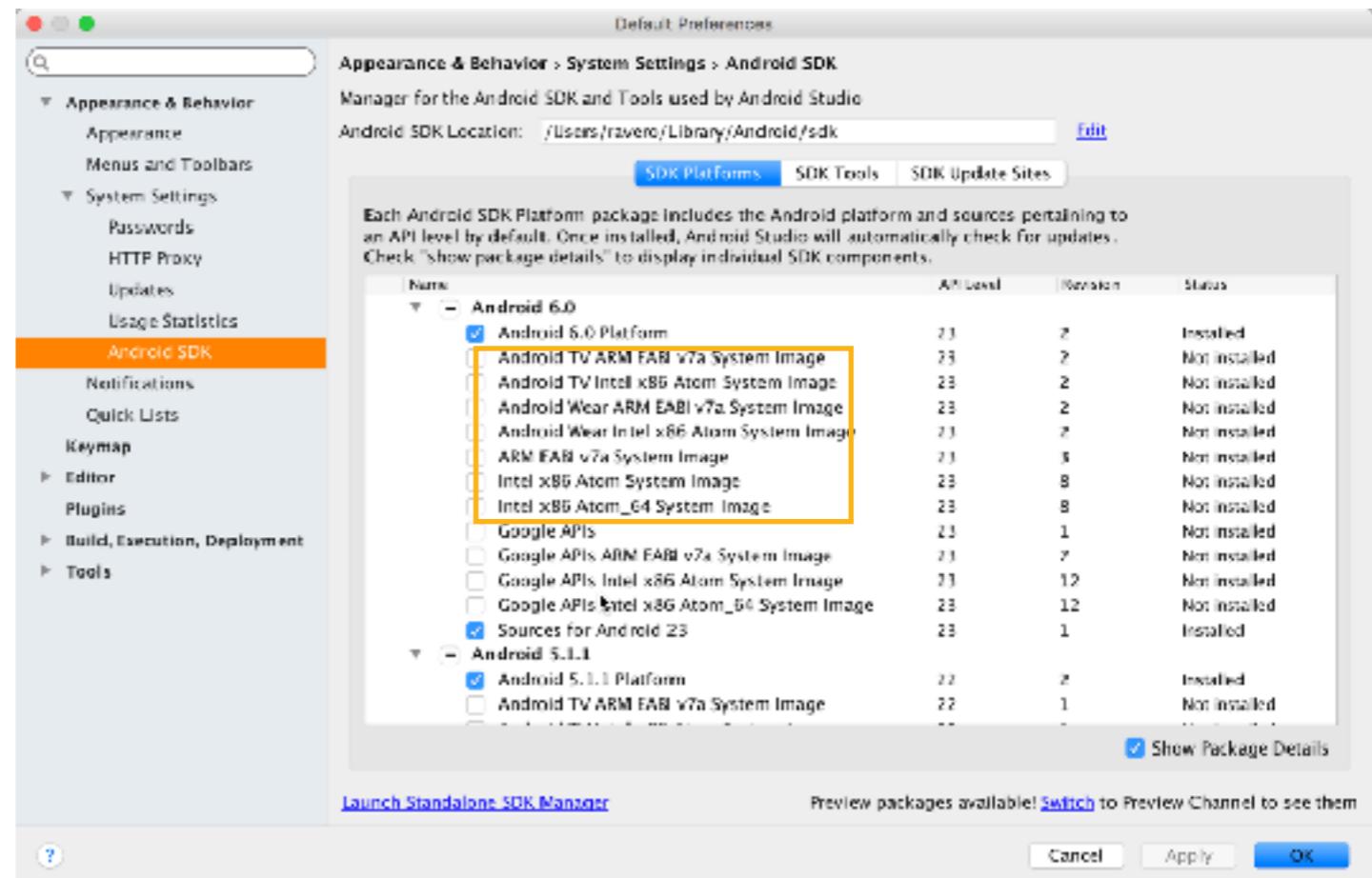
Usando o SDK Manager

Principais Componentes:

System Images

As diversas imagens do sistema operacional para uso com o emulador.

Recomendamos das imagens x86/64 caso o sistema suporte (muito mais rápidas).



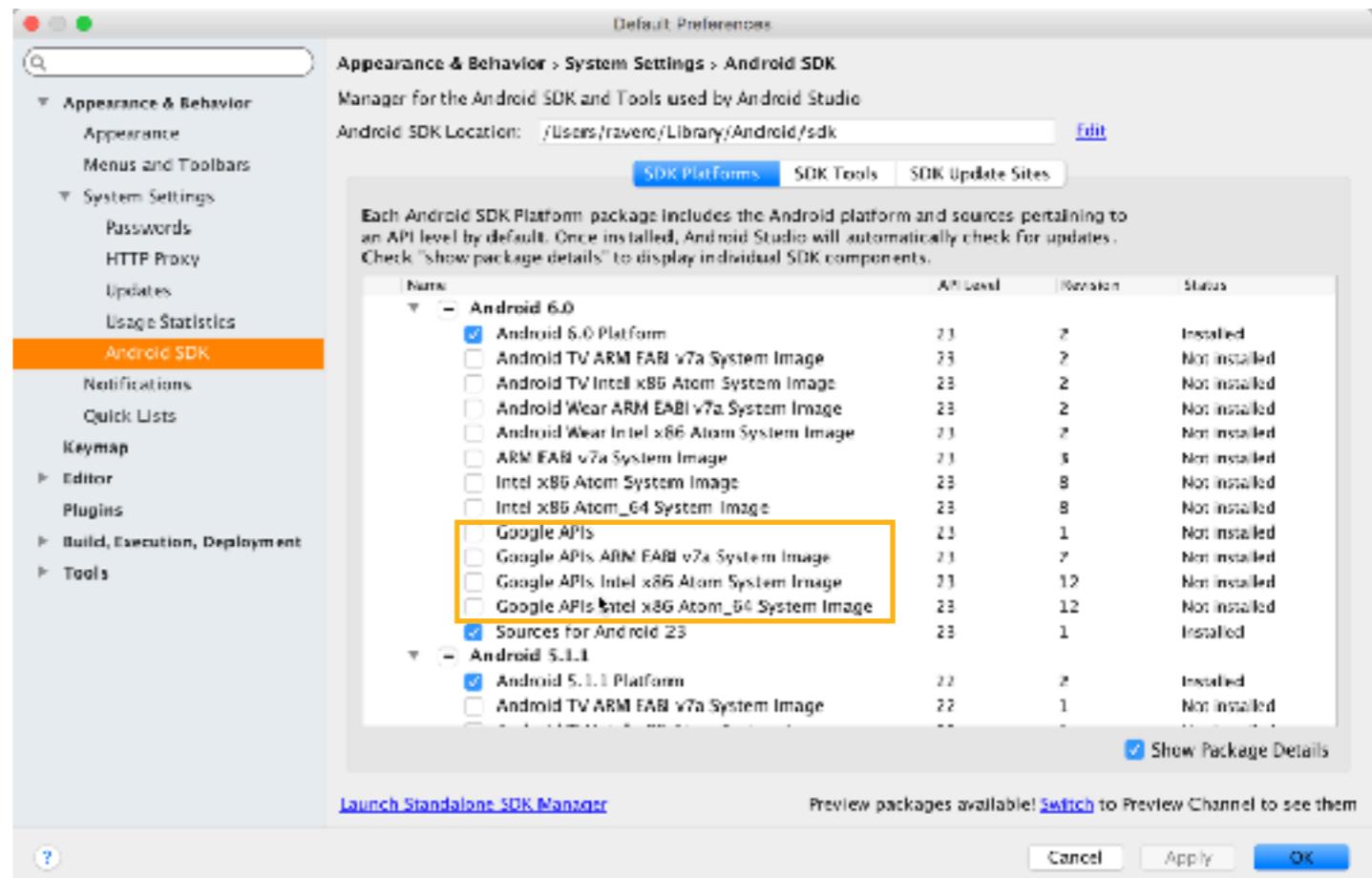
Usando o SDK Manager

Principais Componentes:

Google API's

Imagens do sistema operacional incluindo as API's do Google

Google Search, Google Maps, Gmail, etc...



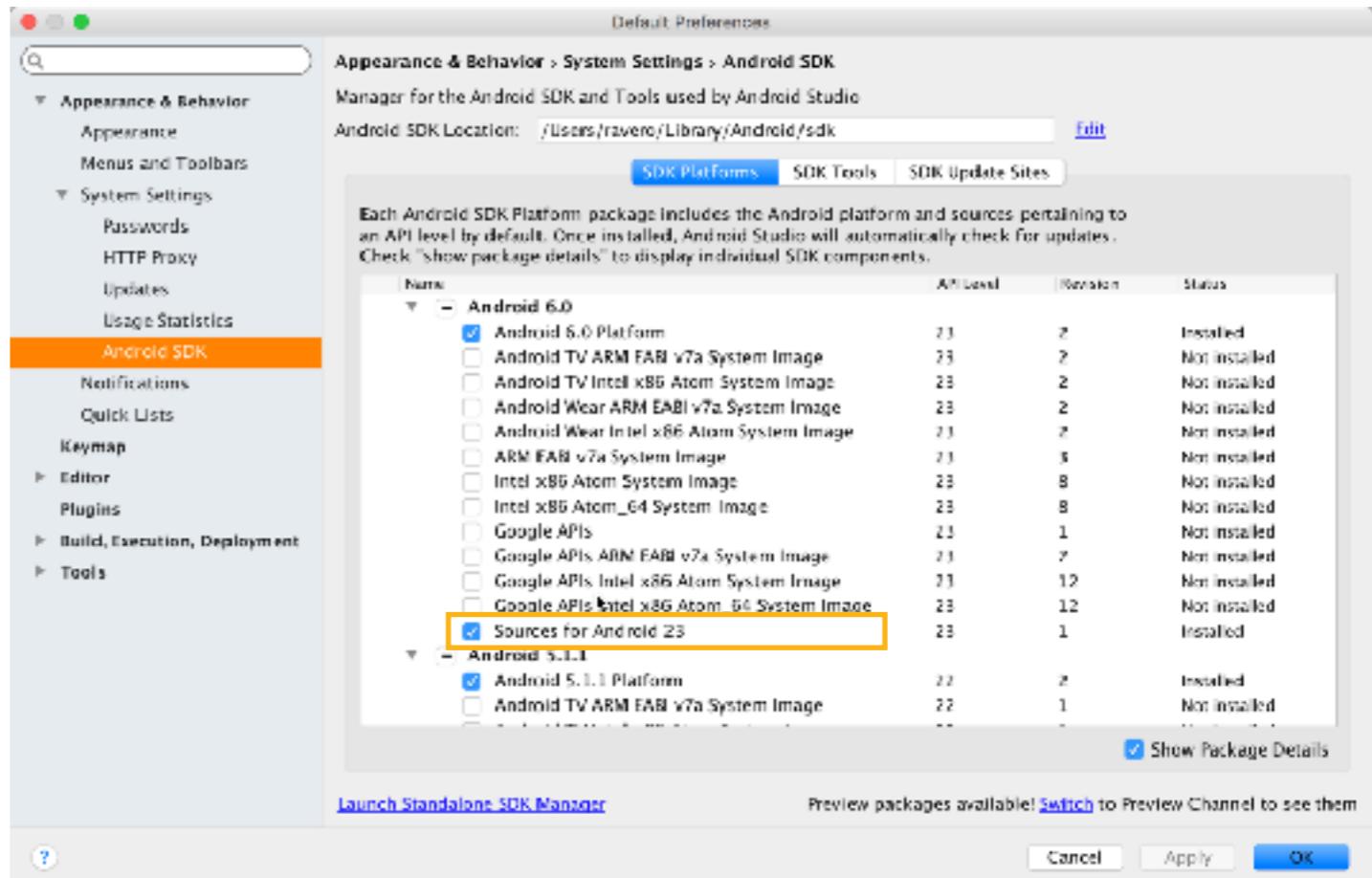
Usando o SDK Manager

Principais Componentes:

Sources

Código fonte do sistema operacional

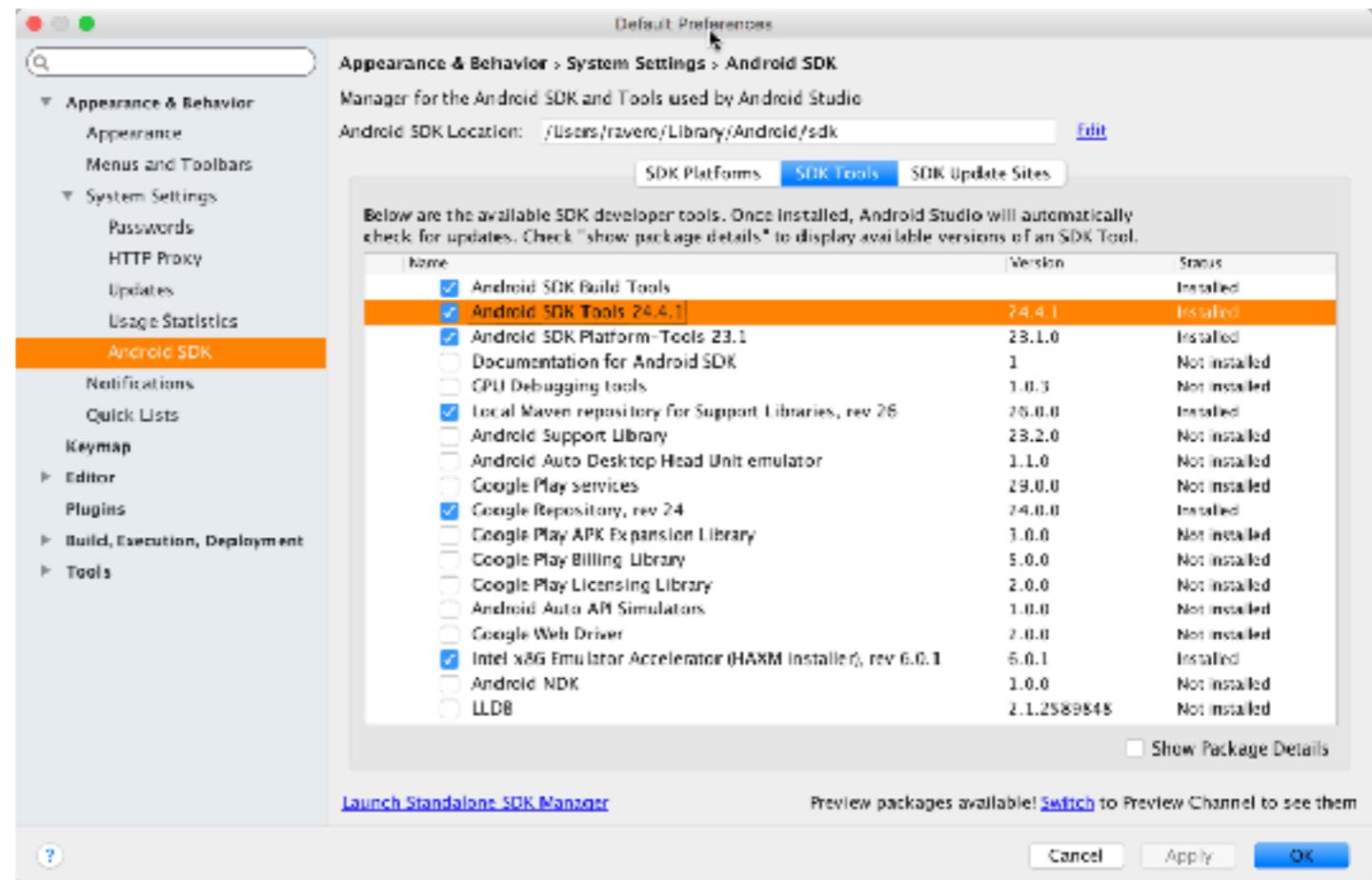
Sua instalação permite acesso direto ao código dos componentes em ambiente de depuração.



Usando o SDK Manager

SDK Tools

Componentes dos API Levels do Android.



Usando o SDK Manager

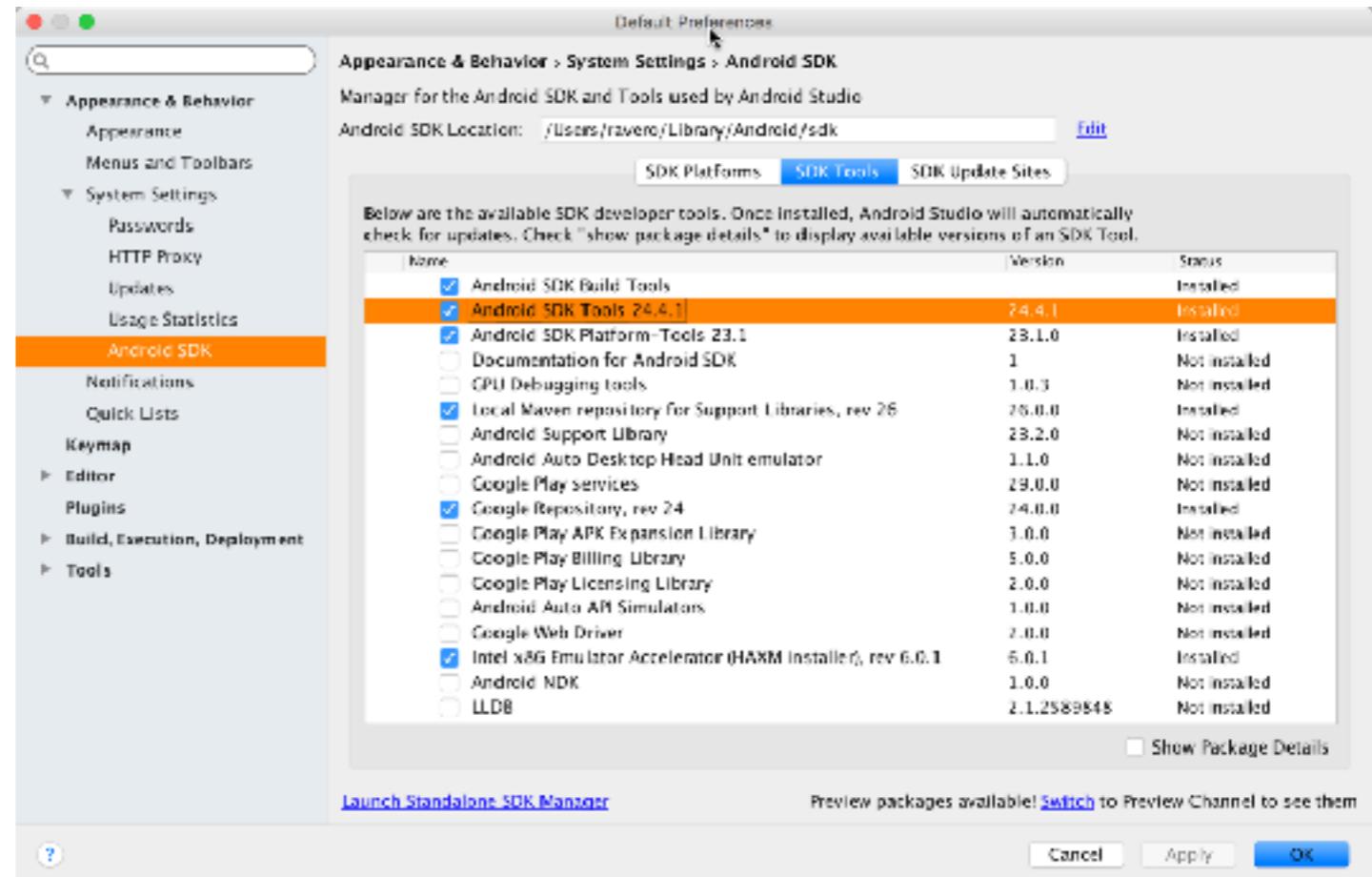
Principais Componentes:

SDK Build Tools

Compiladores do
Android

SDK Tools/Platform Tools

Conjunto de
ferramentas de
desenvolvimento e
debug do Android



4. USANDO O ANDROID STUDIO

Edição de Código
Trabalhando com Resources

Edição de Código

Auto Complete

O Android Studio tenta completar seu código a medida que você digita.

Pode ajudar com:

Nomes de classes

Nomes de variáveis e métodos

Construções de linguagem



Edição de Código

Demonstração (auto complete):

```
dataStore = SQLiteDataStore.getInstance();  
  
// Extrai os extras do Intent para obter o código da Atividade Traqt selecionada  
selectedActivityId = getIntent().getIntExtra(HomeActivity.ACTIVITY_ID_EXTRA, 0);  
selectedActivity = dataStore.getActivityRepository().findById(selectedActivityId);  
if (selectedActivity == null) {  
    setResult(RESULT_CANCELED);  
    finish();  
}
```

I

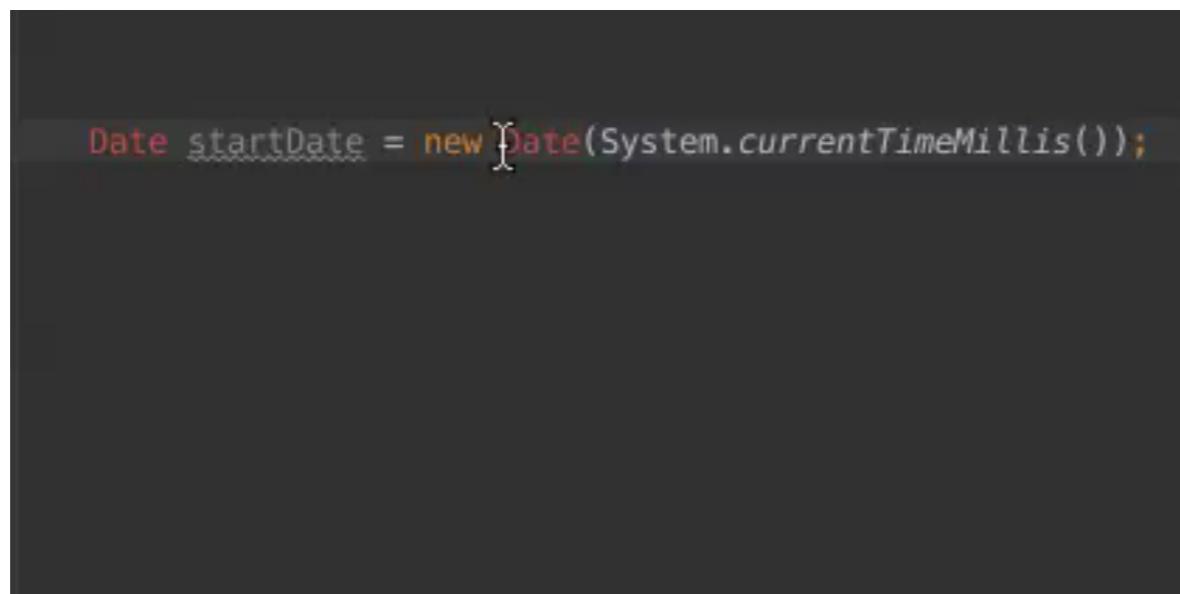
Utilize TAB para completar uma sentença

Edição de Código

Opções

O comando ALT+ENTER traz opções adicionais de acordo com o contexto.

Se o trecho estiver com um erro ou warning ele mostrará opções de correção.



A screenshot of an IDE interface showing a code editor. The code displayed is:

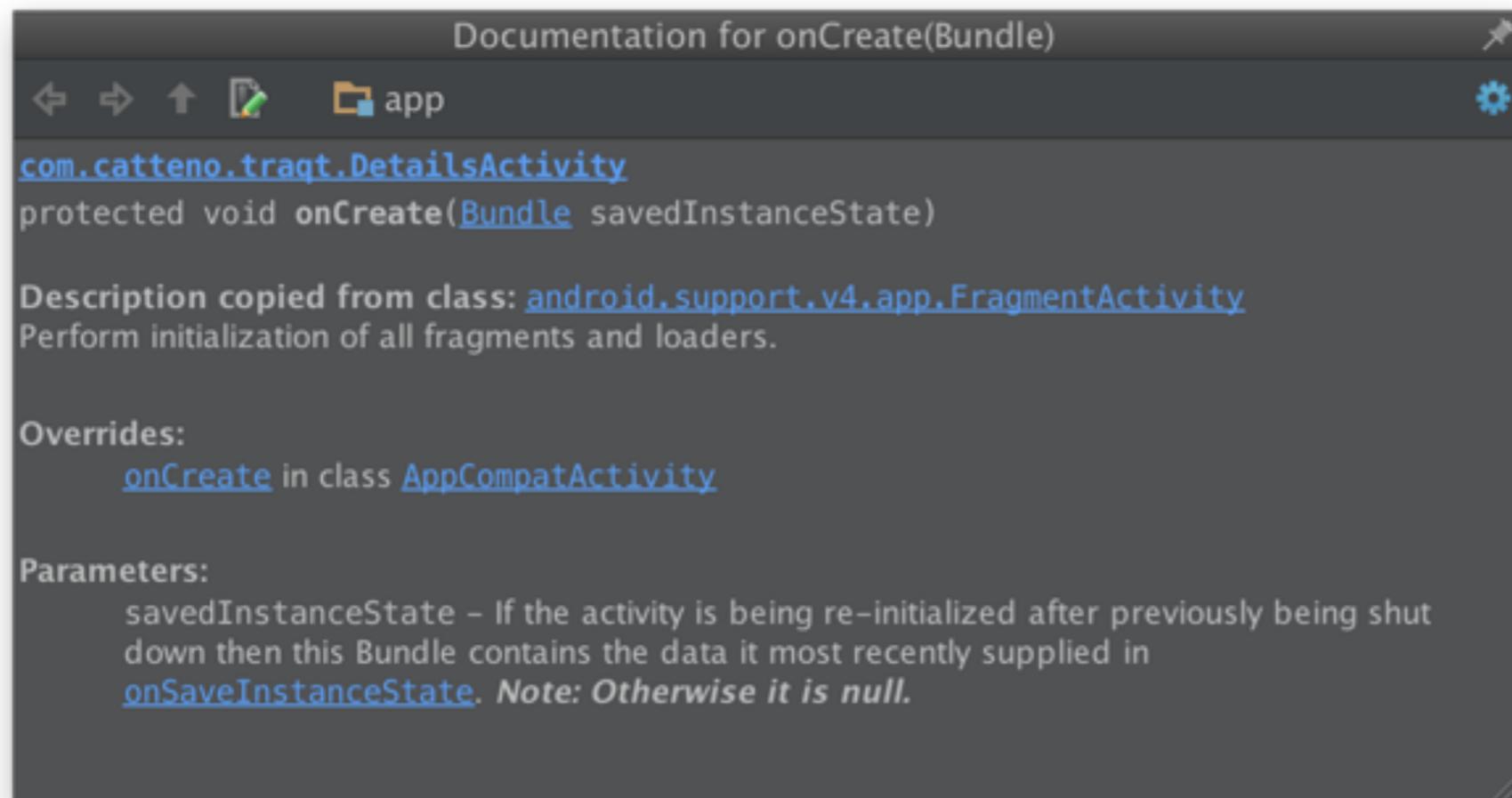
```
Date startDate = new Date(System.currentTimeMillis());
```

The word "Date" is highlighted in red, indicating it is an error or warning. A dropdown menu is open below the cursor, listing several code completion options for "Date".

Edição de Código

Documentação Rápida

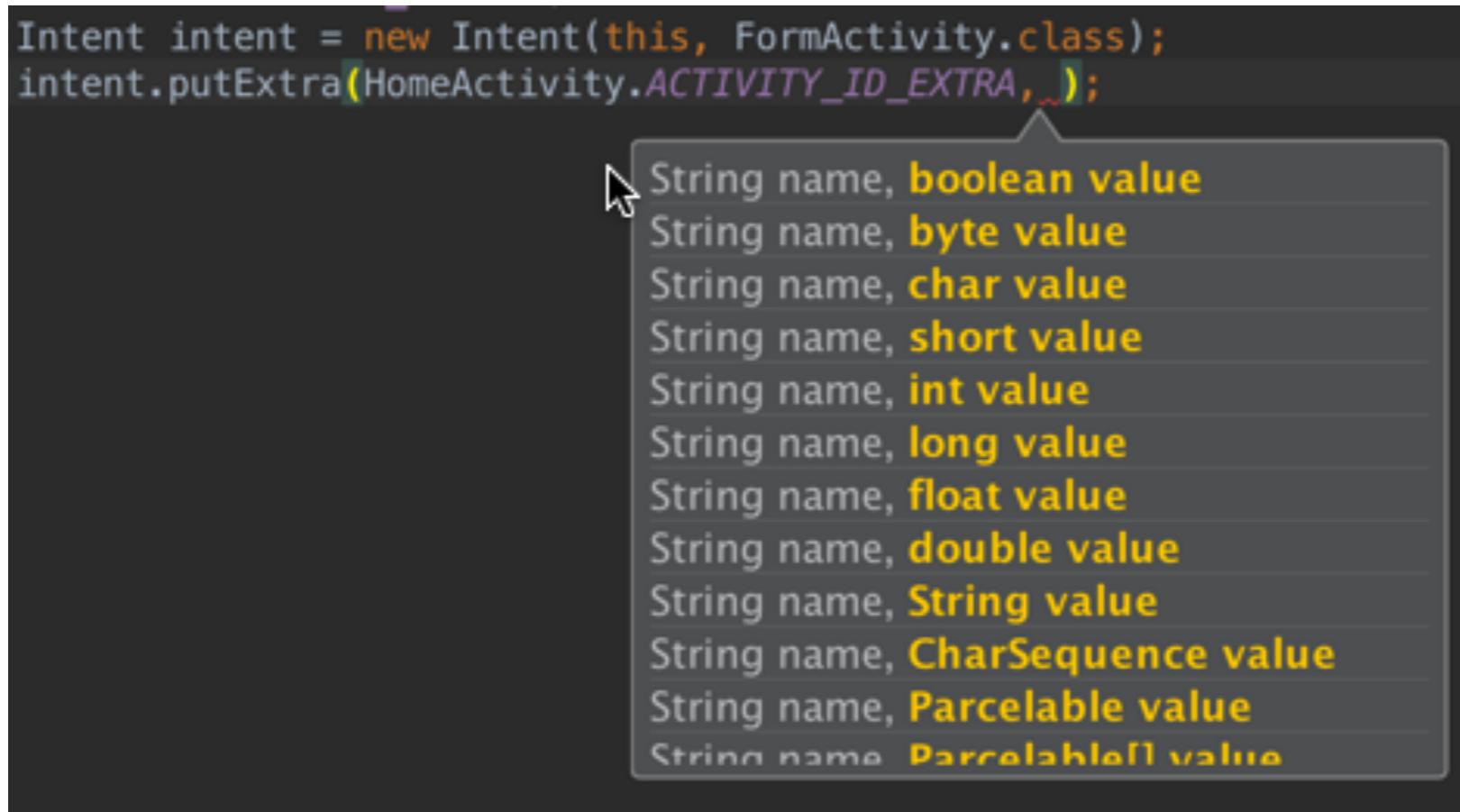
Utilize **CTRL+Q** (Windows) **F1** (Mac) para obter a documentação de uma classe ou método.



Edição de Código

Lista de Parâmetros

Utilize **CTRL+P** (Windows) ou **CMD+P** (Mac) para obter rapidamente a lista de parâmetros de um método.



Edição de Código

Find Usages

Localiza todas as referências em código para um determinado símbolo.

Atalho: ALT+F7

Go To Declaration

Vai para a declaração de um tipo

Caso não tenha o código fonte do tipo mostra seu header.

Atalho: CTRL+B (Windows) ou CMD+B (Mac)



Edição de Código

Ferramentas de Refactoring

O Android Studio traz uma série de ferramentas de Refactoring embutidas

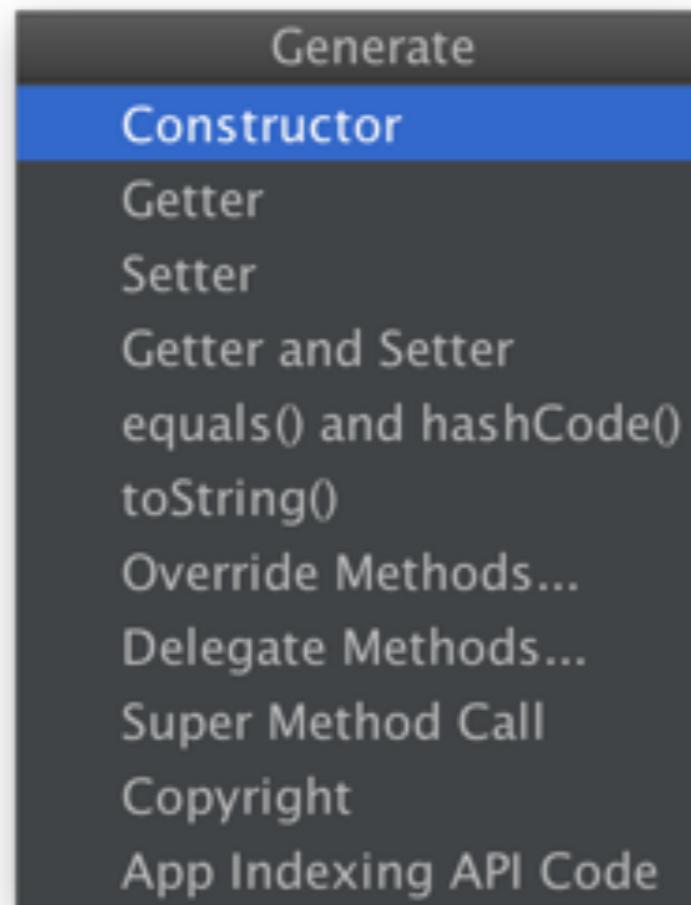
É possível renomear símbolos, extrair métodos, re-ordenar parâmetros, entre outras operações



Edição de Código

Geração rápida de código

Utilizando o atalho ALT+Insert (Windows) ou CMD+N (Mac)
é possível gerar trechos de código rapidamente



Códigos:
Construtores
Getters e/ou Setters
Sobrescrever métodos

Trabalhando com Resources

Editor de Layouts

Android Studio conta com um editor visual de Layouts.

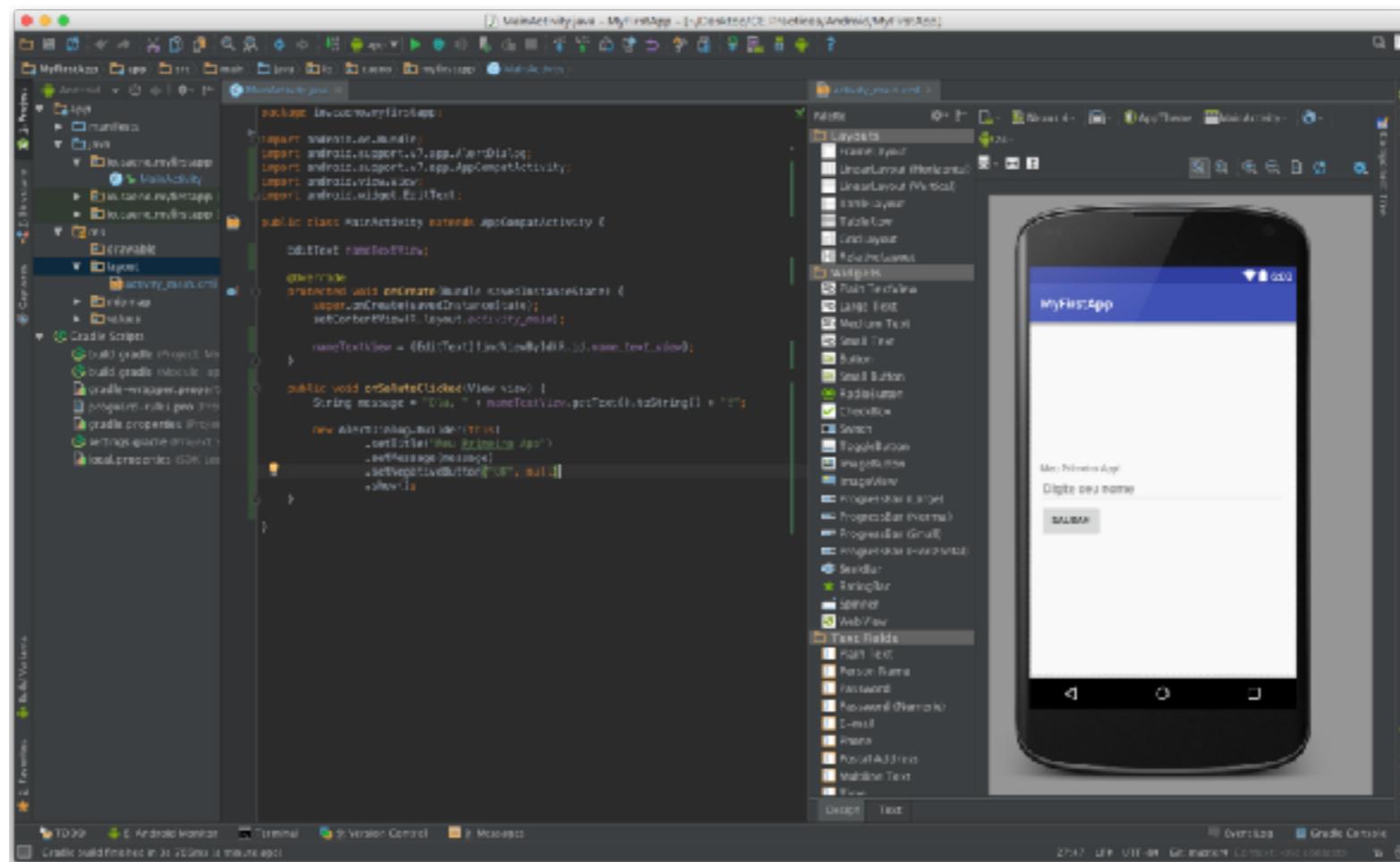
Layouts são arquivos XML que descrevem declarativamente uma interface.

O editor XML conta com um Preview em tempo Real.



Trabalhando com Resources

Editor de Layouts do Android Studio



Trabalhando com Strings

Strings

Projetos Android convencionam centralizar todas as strings em resources.

Isso facilita a tradução e internacionalização de projetos.

Strings declaradas como resources são facilmente acessadas através do código fonte.

5. EMULADOR DE ANDROID

O que são Emuladores?

Prática: Meu primeiro projeto Android

O que são Emuladores?

Emuladores executam as imagens do Sistema Operacional Android dentro de seu computador.

Permitem testar seus Apps sem a necessidade de um dispositivo físico.

Importante para testar em diversas versões do Android em diversos tipos de dispositivos.



Demo

PRÁTICA:

Meu primeiro projeto Android

APRESENTANDO O XCODE

O que é o Android Studio?

Android Developer Tools (ADT)

Instalando o Android Studio

Principais recursos



Xcode

Ferramenta de desenvolvimento oficial para a plataforma Apple (OSX e iOS).

Distribuída gratuitamente pela Apple.

Padrão de Mercado para desenvolvimento iOS.



Xcode

Funcionalidades Integradas:

Gestão de Workspaces e Projetos

Editor de Códigos

Interface Builder

Debugger

Integração com Ferramentas de CVS

Swift Playgrounds





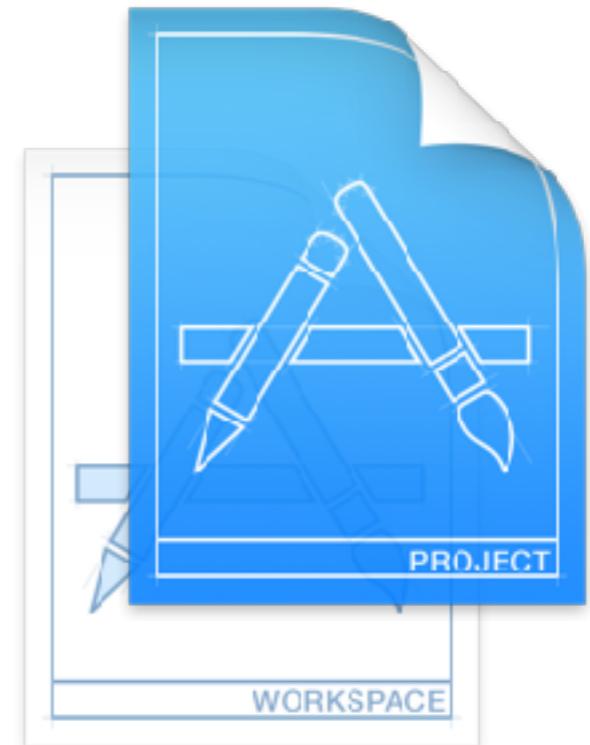
Demo: Swift Playgrounds

Workspaces e Projetos

Workspace: agrupamento lógico de projetos relacionados um mesmo contexto.

Projects: Agrupamento lógico de arquivos e recursos utilizados para produzir um ou mais Targets.

Target: um App, Library ou Extension produzido pela compilação dos códigos e recursos de um projeto.





Demo: Tipos de Projetos e Targets

Conhecendo o Xcode

Partes do Ambiente:

Standard Editor

Navigators

Utilities

Libraries

Assistant Editor

Version Editor

Debug Area



The screenshot shows the Xcode interface with the Standard Editor active. The title bar reads "ViewController.swift". The navigation bar shows the project structure: DemoApp > DemoApp > ViewController.swift. The code editor displays the following Swift code:

```
// ViewController.swift
// DemoApp
//
// Created by Rafael Veronczi on 11/19/14.
// Copyright (c) 2014 Raverco. All rights reserved.

import UIKit

class ViewController: UIViewController {

    // MARK: - Outlets

    @IBOutlet weak var helloLabel: UILabel!

    // MARK: - View Lifecycle

    override func viewDidLoad() {
        super.viewDidLoad()
        // Do any additional setup after loading the view, typically from a nib.
    }

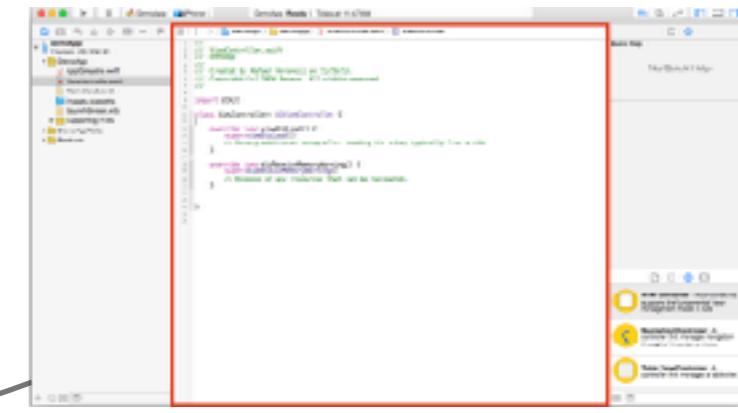
    override func didReceiveMemoryWarning() {
        super.didReceiveMemoryWarning()
        // Dispose of any resources that can be recreated.
    }
}
```

Standard Editor

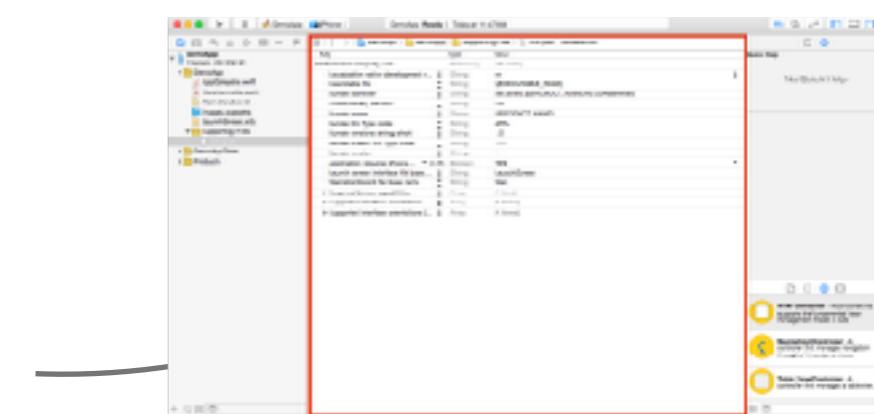
Proporciona um editor apropriado para o tipo de arquivo com que estiver trabalhando:



Código fonte (Swift, Objective-C, outros): Editor de código fonte integrado



Storyboards e XIBs: Interface Builder



Plists: editor de arquivos Plist

Atalho: ⌘ + Enter

Navigators

Proporcionam navegação as diversas partes e recursos do projeto.

Project Navigator (1)

Symbol Navigator (2)

Find Navigator (3)

Issues Navigator (4)

Test Navigator (5)

Debug Navigator (6)

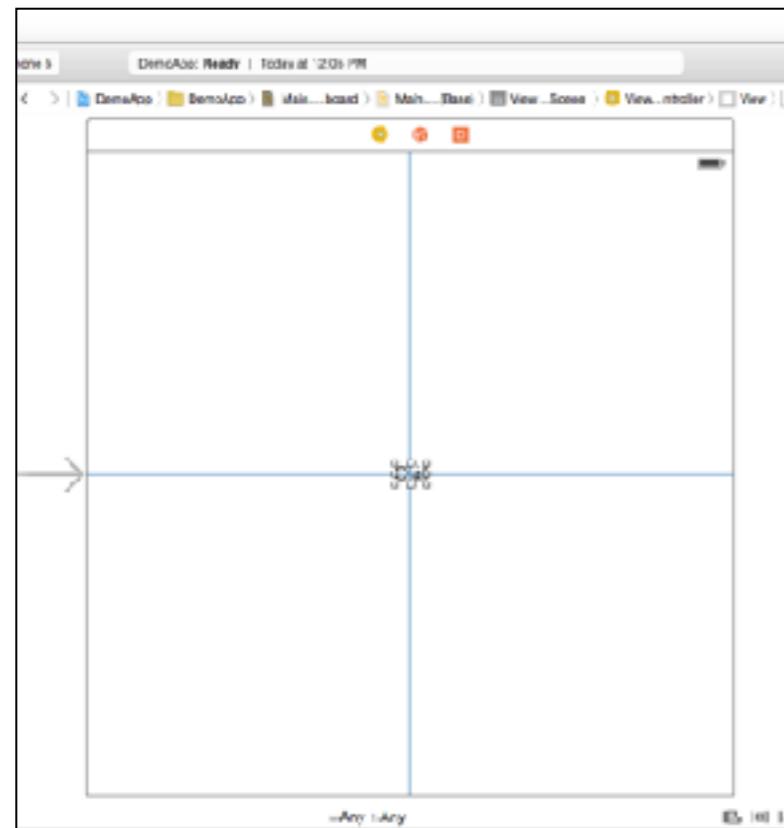
Breakpoints Navigator (7)

Report Navigator (8)

Atalho: ⌘ + n°

Esconder e exibir rapidamente: ⌘ + 0

Utilities



Proporciona acesso a utilitários que permitem configurar e ajustar diversas propriedades e atributos do arquivo sendo editado:

File
Inspector
(1)

Identity
Inspector
(3)

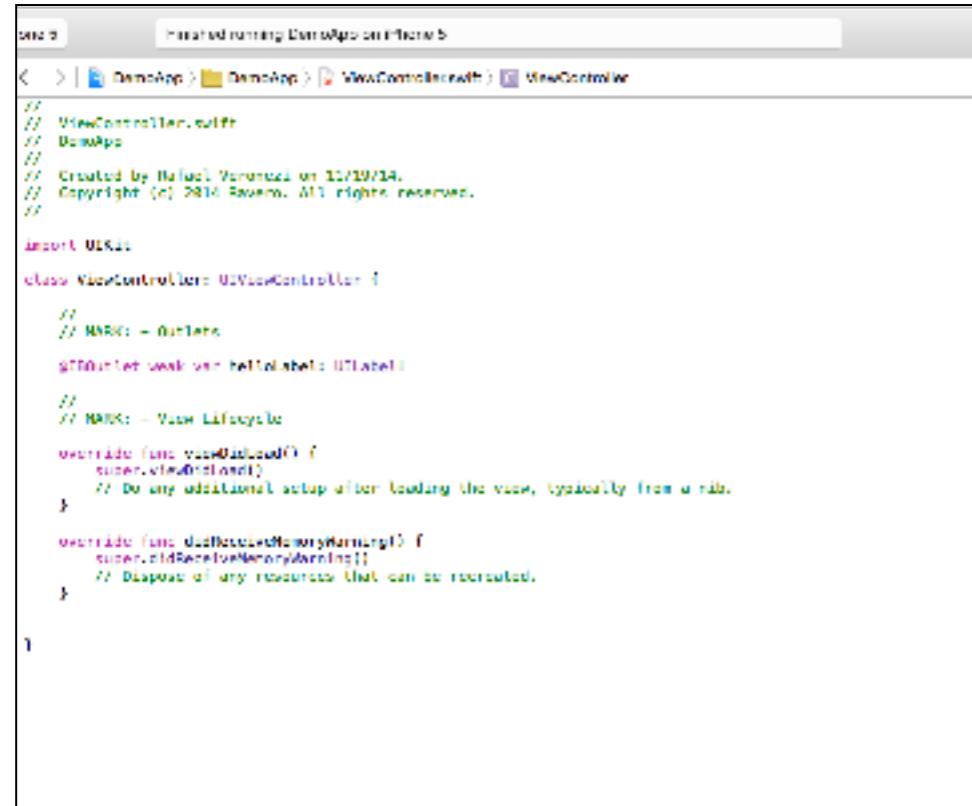
Quick Help
Inspector
(2)

Attributes
Inspector
(4)

Atalho: ⌘ + n°

Esconder e exibir rapidamente: ⌘ + 0

Libraries



A screenshot of the Xcode interface showing the Libraries browser. The sidebar on the left lists categories like 'File', 'Code Snippet', 'Object', and 'Media'. The main area displays code snippets for 'UIViewController+HelloLabel'. The code includes imports, class definitions, and overridden methods for view loading and deallocation.

```
// Viewcontroller.swift
// DemoApp
// Created by Rafael Veronezi on 11/19/14.
// Copyright (c) 2014 Raver. All rights reserved.

import UIKit
class ViewController: UIViewController {
    // MARK: - Outlets
    @IBOutlet weak var helloLabel: UILabel!
    // MARK: - View Lifecycle
    override func viewDidLoad() {
        super.viewDidLoad()
        // Do any additional setup after loading the view, typically from a nib.
    }
    override func didReceiveMemoryWarning() {
        super.didReceiveMemoryWarning()
        // Dispose of any resources that can be recreated.
    }
}
```

Proporciona acesso a bibliotecas de recursos que podem ser usados com os arquivos sendo editados

File Template Library (1)

Code Snippet Library (2)

Object Library (3)

Media Library (4)

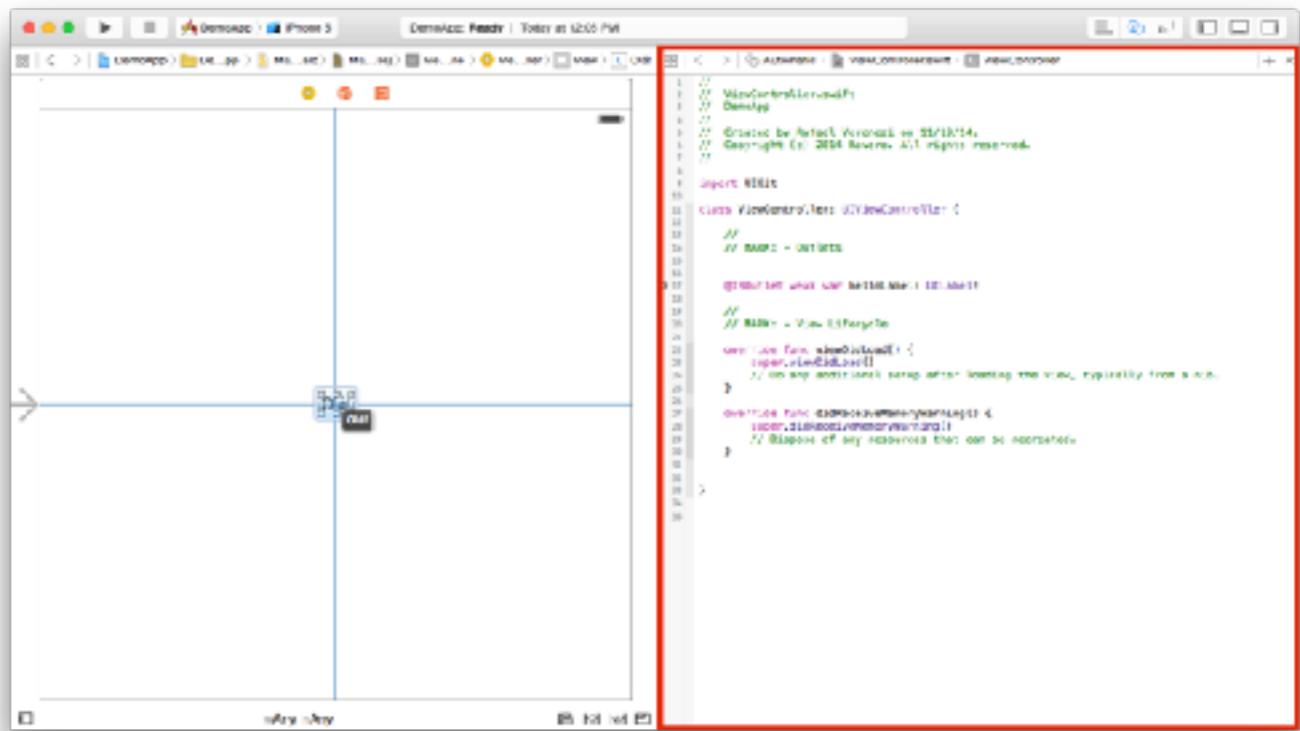
Atalho: ⌘ + Option + Control + n°

Esconder e exibir rapidamente: ⌘ + Option + 0

Assistant Editor

Permite visualizar ou editar arquivos e recursos relacionados ao arquivo aberto no Standard Editor.

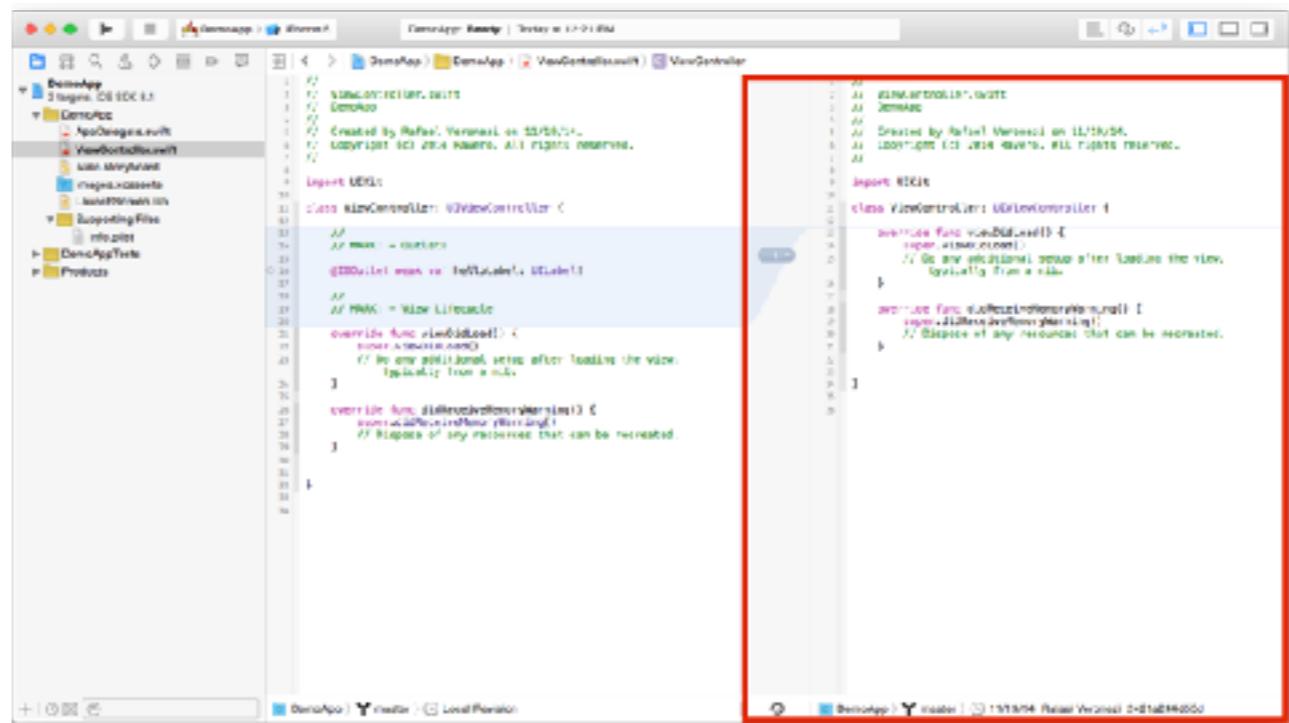
Importante para trabalhar com Interface Builder, para criar conexões entre a Interface e o Código Fonte e visualizar previews dos Designs.



Atalho: ⌘ + Option + Enter

Version Editor

Permite comparar rapidamente o arquivo selecionado no Standard Editor, com versões anteriores registradas no mecanismo de CVS empregado.

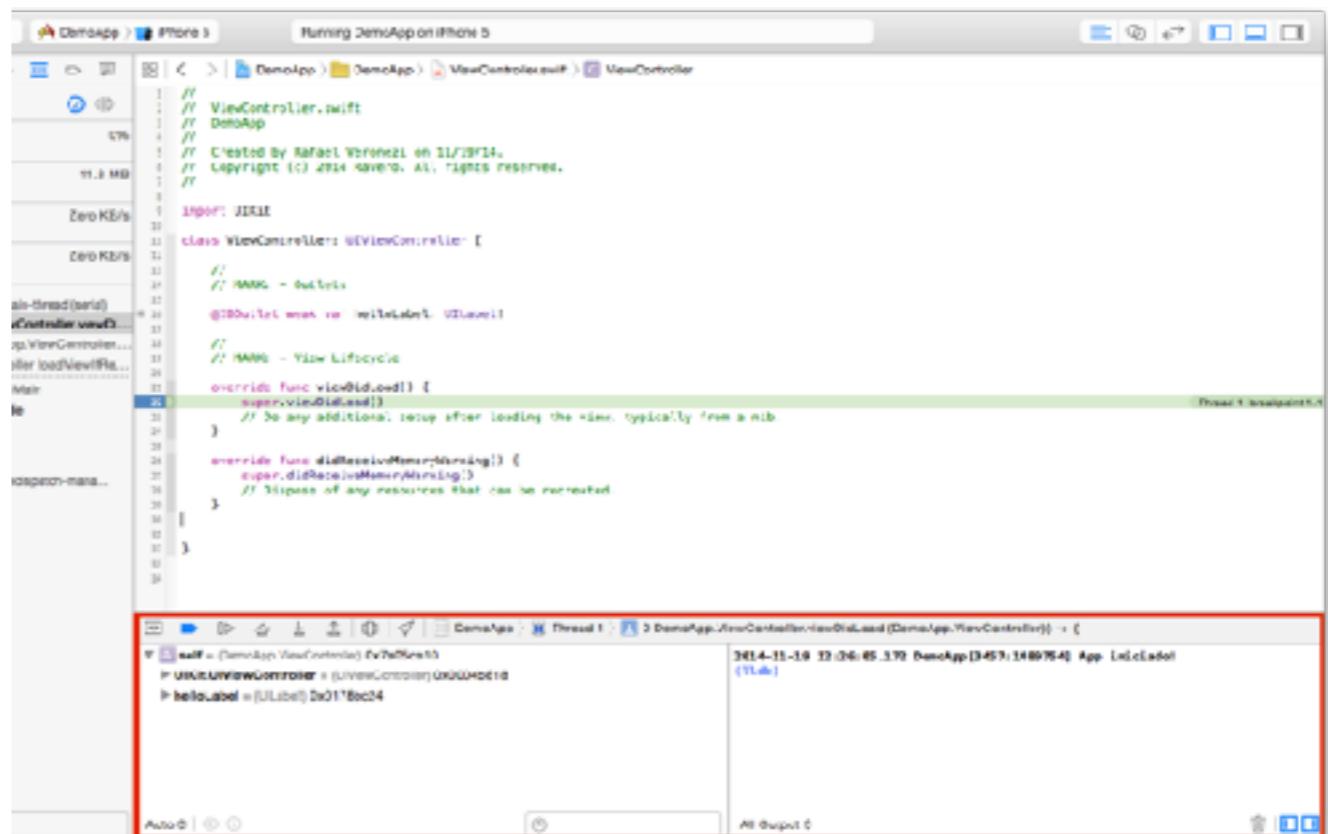


Atalho: ⌘ + Option + Shift + Enter

Debug Area

Exibido quando algum projeto esta sendo executado em modo Debug.

Proporciona Acesso ao Console de Debug e ao Visualizador de Variáveis, que permitem usar funções avançadas de debugging.





Demo: Xcode e iOS Simulator

PRÁTICA:

Meu Primeiro App iOS