

# Android

TechGuide - Alura, FIAP e PM3

---

## Android

### Level 1

#### Kotlin - Fundamentals:

Orientação a Objetos com Kotlin (Properties, Data class, Companion Objects, Delegation)

Kotlin is a cross-platform, object-oriented, functional, statically typed programming language. It compiles to the Java Virtual Machine and can also be translated to the JavaScript language and compiled to native code (via LLVM). It is the official language of Google's Android system.

Understanding its syntax

Knowing the primitive types

Declaring and using variables and constants

Using conditional structures (if, else)

Using repetition structures and loops (while, for)

Using functions, passing parameters and arguments

Implementing methods and reusing them

Null Safety (Eliminate the danger of null references)

Exceptions and Throwables

Coding conventions

Manipulating Collections, arrays and lists

Functional Paradigm Features

OOP with Kotlin (Properties, Data class, Companion Objects, Delegation)

#### Contents

WebSite Kotlin Documentation: Get started with Kotlin (<https://kotlinlang.org/docs/getting-started.html>)

WebSite Kotlin Documentation: Basic types in Kotlin (<https://kotlinlang.org/docs/basic-types.html>)

WebSite Kotlin Documentation: Basic syntax in Kotlin (<https://kotlinlang.org/docs/basic-syntax.html>)

WebSite Kotlin Documentation: Functions in Kotlin (<https://kotlinlang.org/docs/functions.html>)

WebSite Kotlin Documentation: Coding conventions in Kotlin (<https://kotlinlang.org/docs/coding-conventions.html>)

WebSiteGeeksForGeeks: Kotlin Collections (<https://www.geeksforgeeks.org/kotlin-collections/>)

WebSiteGeeksForGeeks: Kotlin Null Safety (<https://www.geeksforgeeks.org/kotlin-null-safety/>)  
Article The Power of Kotlin's Null <https://medium.com/@android-blaze/the-power-of-kotlins-null-safety-fe9cba2a5d5d> )  
YouTubefreeCodeCamp.org: Kotlin Course - Tutorial for Beginners (<https://youtu.be/F9UC9DY-vIU>)  
YouTube freeCodeCamp.org: Learn Kotlin Programming – Full <https://www.youtube.com/watch?v=EExSSotojVI> )  
Course for Beginners (

## Android - Fundamentals:

Getting to know Kotlin, Java or C++, which are the languages to develop Android apps.  
Understanding how the Android SDK packages the App code and resources into an APK (Android Package) to run on the Android OS  
Knowing the input components of an App - Activity, Service, Broadcast Receiver and Content Provider  
Understanding the Android component life cycle and how it works - The Activity life cycle  
Activating App input components with Intents  
Knowing the manifest file and the main configuration items  
Understanding what the resources of an Android project are - source code, static resources, drawables, layout, mipmap, values etc  
Creating an Android project with Android Studio and running it on a physical or virtual device  
Knowing the jetpack libraries to ensure compatibility between Android versions

## Contents

WebSite Android Documentation: <https://developer.android.com/guide/components/fundamentals>  
Application Fundamentals ( )  
Android  
WebSite Documentation: <https://developer.android.com/guide/components/fundamentals#Components>  
App components ( )  
Android  
WebSite Documentation: <https://developer.android.com/guide/components/fundamentals#ActivatingComponents>  
Activating components ( )  
WebSite Android Documentation: App <https://developer.android.com/guide/topics/manifest/manifest-intro>  
Manifest Overview ( )  
WebSite Android Documentation: <https://developer.android.com/guide/topics/resources/providing-resources>  
App resources overview ( )  
WebSite Android Documentation: Build your <https://developer.android.com/training/basics/firstapp>  
first Android app ( )  
WebSite Android Documentation: Explore <https://developer.android.com/jetpack/androidx/explorer>  
the Jetpack libraries by type ( )  
YouTube freeCodeCamp.org: Android Development for <https://www.youtube.com/watch?v=fis26HvvDI> )  
Beginners - Full Course (

## Object-oriented Programming Concepts:

Object-oriented programming (OOP) is a programming paradigm based on the concept of 'objects', which can contain data and code: data in the form of fields (often known as attributes or properties), and code, in the form of procedures (often known as methods). A common feature of objects is that procedures (or methods) are attached to them and can access and modify the object's data fields. Some of the main concepts are classes and instances, inheritance, and encapsulation.

How objects work

Creating and using constructors

What classes are

Creating and using Methods

How encapsulation works

What inheritance is

What polymorphism is

How interfaces work

What abstractions are

### Contents

WebSite	MDN Web Docs: Object-oriented programming (	<a href="https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Objects/Object-oriented_programming">https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Objects/Object-oriented_programming</a>	)
WebSite	Geeks for Geeks: Object Oriented Programming (OOPs) Concept in Java (	<a href="https://www.geeksforgeeks.org/object-oriented-programming-oops-concept-in-java/">https://www.geeksforgeeks.org/object-oriented-programming-oops-concept-in-java/</a>	)
WebSite	W3Schools: Object-Oriented Programming Style (	<a href="https://www.w3schools.blog/object-oriented-programming-style">https://www.w3schools.blog/object-oriented-programming-style</a>	)
Article	What is object-oriented programming? OOP explained in depth (	<a href="https://www.educative.io/blog/object-oriented-programming">https://www.educative.io/blog/object-oriented-programming</a>	)
YouTube	Programming with Mosh: Object-oriented Programming in 7 minutes (	<a href="https://www.youtube.com/watch?v=QJjY2TNyl-8">https://www.youtube.com/watch?v=QJjY2TNyl-8</a>	)
YouTube	freeCodeCamp.org: Intro to Object Oriented Programming - Crash Course (	<a href="https://www.youtube.com/watch?v=SiBw7os-zI">https://www.youtube.com/watch?v=SiBw7os-zI</a>	)
YouTube	Traversy Media: JavaScript OOP Crash Course (ES5 & ES6) (	<a href="https://www.youtube.com/watch?v=vDJpGenyHaA">https://www.youtube.com/watch?v=vDJpGenyHaA</a>	)

## Android - View system:

The View class represents the basic building block of user interface components in Android.

Knowing the View and ViewGroup reference for defining a Layout

Knowing the main ViewGroups to build layouts - LinearLayout, RelativeLayout, FrameLayout and ConstraintLayout

Using the visual Layout editor to customize the layout, or use the XML file for editing

Knowing the Preview of the visual editor and use it to prepare the screens and components of the App

Using the Views from the Android SDK - TextView, Button, EditText, CheckBox, DatePicker, etc

Creating custom Views to meet specific demands

Customizing Views with Android or View properties - height, width, visibility, spacing adjustments

Implementing dynamic content layouts with adapters - AdapterView and RecyclerView

Integrating business code with screen layout - Calling and manipulating Views with View Binding

Reacting to View events from listeners such as clicks, long clicks, scrolling, drag and drop

## Contents

WebSite	Android Documentation: <a href="https://developer.android.com/develop/ui/views/layout/declaring-layout">https://developer.android.com/develop/ui/views/layout/declaring-</a> Layouts ( <a href="https://developer.android.com/develop/ui/views/layout/declaring-layout">layout</a> )
WebSite	Android Documentation: <a href="https://developer.android.com/develop/ui/views/layout/declaring-layout#CommonLayouts">https://developer.android.com/develop/ui/views/layout/declaring-</a> Common Layouts ( <a href="https://developer.android.com/develop/ui/views/layout/declaring-layout#CommonLayouts">layout#CommonLayouts</a> )
WebSite	Android Documentation: Build a UI with <a href="https://developer.android.com/studio/write/layout-editor">https://developer.android.com/studio/write/layout-</a> Views - Introduction to the Layout Editor ( <a href="https://developer.android.com/studio/write/layout-editor">editor</a> )
WebSite	Android Documentation: <a href="https://developer.android.com/develop/ui/views/layout/constraint-layout">https://developer.android.com/develop/ui/views/layout/constraint-</a> Build a Responsive UI ( <a href="https://developer.android.com/develop/ui/views/layout/constraint-layout">layout</a> )
WebSite	GeeksForGeeks: Android View <a href="https://www.geeksforgeeks.org/android-view-hierarchy">https://www.geeksforgeeks.org/android-view-</a> Hierarchy ( <a href="https://www.geeksforgeeks.org/android-view-hierarchy">hierarchy</a> )
Article	Introduction to Android Views and <a href="https://www.studytonight.com/android/introduction-to-views">https://www.studytonight.com/android/introduction-</a> ViewGroups ( <a href="https://www.studytonight.com/android/introduction-to-views">to-views</a> )
Article	Designing complex UI using <a href="https://proandroiddev.com/designing-complex-ui-using-android-constraintlayout-cb0606823da0">https://proandroiddev.com/designing-complex-ui-using-</a> Android ConstraintLayout ( <a href="https://proandroiddev.com/designing-complex-ui-using-android-constraintlayout-cb0606823da0">android-constraintlayout-cb0606823da0</a> )
Article	Listas com <a href="https://medium.com/android-dev-br/listas-com-recyclerview-d3f41e0d653c">https://medium.com/android-dev-br/listas-com-recyclerview-</a> RecyclerView ( <a href="https://medium.com/android-dev-br/listas-com-recyclerview-d3f41e0d653c">d3f41e0d653c</a> )
WebSite	Android Documentation: Input <a href="https://developer.android.com/develop/ui/views/touch-and-input/input-events">https://developer.android.com/develop/ui/views/touch-</a> events overview ( <a href="https://developer.android.com/develop/ui/views/touch-and-input/input-events">and-input/input-events</a> )
YouTube	Coding Pursuit: View, View Groups and Layouts in <a href="https://www.youtube.com/watch?v=TPdXdXr1ghU">https://www.youtube.com/watch?</a> Android ( <a href="https://www.youtube.com/watch?v=TPdXdXr1ghU">v=TPdXdXr1ghU</a> )

## Android - Fragments:

A Fragment represents the behavior or a part of the user interface in a host Activity. You can combine multiple fragments into a single Activity to create a multi-panel UI and reuse a fragment in multiple activities. You can imagine a fragment as a modular section of an Activity, which has its own life cycle, receives its own input events and can be added or removed during the execution of the Activity.

Understanding what a Fragment is

How to use and reuse Fragments in the same Activity

Understanding the reasons to use Fragments in Android projects

Implementing layouts with multiple panels

Migrating Android projects that only use Activities to use Fragments

Handling transactions from the fragment manager

## Contents

WebSite	Android Documentation: Fragments (	<a href="https://developer.android.com/guide/fragments">https://developer.android.com/guide/fragments</a> )
WebSite	Android Documentation: Fragment lifecycle (	<a href="https://developer.android.com/guide/fragments/lifecycle">https://developer.android.com/guide/fragments/lifecycle</a> )
WebSite	Android Documentation: Create a fragment (	<a href="https://developer.android.com/guide/fragments/create">https://developer.android.com/guide/fragments/create</a> )
Article	Android Fragments: Common Queries & Mistakes (	<a href="https://betterprogramming.pub/android-fragments-common-queries-mistakes-1c42e9f6b44f">https://betterprogramming.pub/android-fragments-common-queries-mistakes-1c42e9f6b44f</a> )
YouTube	Philipp Lackner: FRAGMENTS - Android Fundamentals (	<a href="https://www.youtube.com/watch?v=vAl7RSPxOA">https://www.youtube.com/watch?v=vAl7RSPxOA</a> )

## Android - App navigation:

Screen navigation in apps is fundamental to the user experience of Android Apps. For this it is essential to know the principles of Android navigation.

Learning what the Navigation library is and how you can implement it, either for the Views system or for Jetpack Compose

Integrating navigation with app components, either in tabbed navigation, or in visual component views depending on the screen

Knowing what an Android back stack task is

Knowing the app links - Deep Link, Web Links, and Android App Links

## Contents

WebSite	Android Documentation: Principles of navigation (	<a href="https://developer.android.com/guide/navigation/navigation-principles">https://developer.android.com/guide/navigation/navigation-principles</a> )
WebSite	Android Documentation: Navigation (	<a href="https://developer.android.com/guide/navigation?hl=pt-br">https://developer.android.com/guide/navigation?hl=pt-br</a> )
WebSite	Android Documentation: Navigating with Compose (	<a href="https://developer.android.com/jetpack/compose/navigation">https://developer.android.com/jetpack/compose/navigation</a> )
WebSite	Android Documentation: Tasks and the back stack (	<a href="https://developer.android.com/guide/components/activities/tasks-and-back-stack">https://developer.android.com/guide/components/activities/tasks-and-back-stack</a> )
Article	Understand Jetpack Compose Navigation in 3 Minutes (	<a href="https://medium.com/@cybercoder.naj/compose-navigation-in-3-minutes-5cff3c57c34e">https://medium.com/@cybercoder.naj/compose-navigation-in-3-minutes-5cff3c57c34e</a> )
YouTube	Android Developers: Android Jetpack - Introducing Navigation component (	<a href="https://www.youtube.com/watch?v=YOCs2MQxyls">https://www.youtube.com/watch?v=YOCs2MQxyls</a> )
YouTube	Coding in Flow: Getting Started With Navigation Component (	<a href="https://www.youtube.com/playlist?list=PLrnPJCHvNZuCamMFswP597mUF-whXoAA6">https://www.youtube.com/playlist?list=PLrnPJCHvNZuCamMFswP597mUF-whXoAA6</a> )

## Jetpack Compose - Fundamentals:

Jetpack Compose is a tool that brings the proposal to create native Android interfaces with less code, faster and more beautiful your apps, it does this through the declarative approach.

Creating an Android app from scratch using Jetpack Compose

Building Layouts from Composables

Previewing Composables

Managing states, events, composing and recomposing

Configuring Compose in an existing project and applying interoperability

Working with forms

Understanding the difference between XML and Compose

Maintaining states using the MVVM pattern with ViewModel and StateFlow, understanding the life cycle

## Contents

WebSite	Android Documentation: Jetpack Compose Tutorial (	<a href="https://developer.android.com/jetpack/compose/tutorial">https://developer.android.com/jetpack/compose/tutorial</a> )
Article	Should new Android developers learn Compose or XML? (	<a href="https://dev.to/aldok/should-new-android-developer-learn-compose-or-xml-3oah">https://dev.to/aldok/should-new-android-developer-learn-compose-or-xml-3oah</a> )
WebSite	Android Documentation: State and Jetpack Compose (	<a href="https://developer.android.com/jetpack/compose/state">https://developer.android.com/jetpack/compose/state</a> )
WebSite	Android Documentation: Lifecycle of composables (	<a href="https://developer.android.com/jetpack/compose/lifecycle">https://developer.android.com/jetpack/compose/lifecycle</a> )
YouTube	Philipp Lackner: The Jetpack Compose Beginner Crash Course (	<a href="https://www.youtube.com/watch?v=6_wK_Ud8--0">https://www.youtube.com/watch?v=6_wK_Ud8--0</a> )
YouTube	tutorialsEU: Free Android Jetpack Course (	<a href="https://www.youtube.com/watch?v=-TWz30owX_I">https://www.youtube.com/watch?v=-TWz30owX_I</a> )

## Android - Persistence:

The concept of "data persistence" refers to ensuring that the information inputted into an application will be stored in a medium where it can be retrieved consistently. In other words, they are permanent records that are not lost when the session is closed.

In Android we can persistently store App-specific or shared files, primitive type information with preferences, and structured database data.

## Contents

WebSite	Android Documentation: Data and file storage overview (	<a href="https://developer.android.com/training/data-storage">https://developer.android.com/training/data-storage</a> )
WebSite	Android Documentation: Access app-specific files (	<a href="https://developer.android.com/training/data-storage/app-specific">https://developer.android.com/training/data-storage/app-specific</a> )
WebSite	Android Documentation: Overview of shared storage (	<a href="https://developer.android.com/training/data-storage/shared">https://developer.android.com/training/data-storage/shared</a> )
WebSite	Android Documentation: Save key-value data (	<a href="https://developer.android.com/training/data-storage/shared-preferences">https://developer.android.com/training/data-storage/shared-preferences</a> )
WebSite	Android Documentation: Save data in a local database using Room (	<a href="https://developer.android.com/training/data-storage/room">https://developer.android.com/training/data-storage/room</a> )

WebSite	GeeksForGeeks: Introduction to Room Persistent Library in Android (	<a href="https://www.geeksforgeeks.org/introduction-to-room-persistent-library-in-android/">https://www.geeksforgeeks.org/introduction-to-room-persistent-library-in-android/</a>	)
Article	A Beginner's Guide to the Room Persistence Library (	<a href="https://medium.com/free-code-camp/room-sqlite-beginner-tutorial-2e725e47bfab">https://medium.com/free-code-camp/room-sqlite-beginner-tutorial-2e725e47bfab</a>	)
YouTube	Coding in Flow: Data Persistence - Android Programming (	<a href="https://www.youtube.com/playlist?list=PLrnPJCHvNZuDJsRLWnl-AV4WQVgkJ1HU4">https://www.youtube.com/playlist?list=PLrnPJCHvNZuDJsRLWnl-AV4WQVgkJ1HU4</a>	)

## Android - Gradle:

Gradle and the Android plug-in for Gradle(AGP) provide a flexible way to build, build, manage, and package your Android app or library.

What is a build tool

Learning the structure of an Android project as a multi-module Gradle project

What is Gradle for and how to use it

What are dependencies and how to use them

### Contents

WebSite	Configure your build (	<a href="https://developer.android.com/studio/build">https://developer.android.com/studio/build</a> )	
WebSite	Gradle tips and recipes (	<a href="https://developer.android.com/studio/build/gradle-tips">https://developer.android.com/studio/build/gradle-tips</a> )	
WebSite	Add build dependencies (	<a href="https://developer.android.com/studio/build/dependencies">https://developer.android.com/studio/build/dependencies</a> )	
WebSite	Documentação Android: Guide to Android app modularization (	<a href="https://developer.android.com/topic/modularization">https://developer.android.com/topic/modularization</a> )	
WebSite	GeeksForGeeks: Android   build.gradle (	<a href="https://www.geeksforgeeks.org/android-build-gradle/">https://www.geeksforgeeks.org/android-build-gradle/</a>	)
Article	Intro to App Modularization (	<a href="https://proandroiddev.com/intro-to-app-modularization-42411e4c421e">https://proandroiddev.com/intro-to-app-modularization-42411e4c421e</a>	)
YouTube	Gradle: First Steps with Android Studio and Gradle (	<a href="https://www.youtube.com/watch?v=ieeD3Dc5QjM">https://www.youtube.com/watch?v=ieeD3Dc5QjM</a>	)
YouTube	Philipp Lackner: Gradle for Beginners (Build Types, Product Flavors, Build Variants, Source Sets) (	<a href="https://www.youtube.com/watch?v=o0M4f5djJTQ">https://www.youtube.com/watch?v=o0M4f5djJTQ</a>	)
WebSite	Slides about Android Development - by Moro (	<a href="https://github.com/gabrielbmoro/slides-about-android-development/blob/main/README.md#gradle-">https://github.com/gabrielbmoro/slides-about-android-development/blob/main/README.md#gradle-</a>	)

## Android - Image loading:

Images come in different sizes and shapes. In most cases, they are larger than needed for the UI of an app. Since we are working with limited memory, Android uses a bitmap decoding technique, to avoid all the work of using this we have several libraries that facilitate this process.

What is Bitmap

Knowing the image loading libraries (Glide, Picasso, Coil, and others) and using them

Advantages and disadvantages of the image loading libraries

### Contents

WebSite	TutorialsPoint: What is Bitmap? (	<a href="https://www.tutorialspoint.com/What-is-Bitmap">https://www.tutorialspoint.com/What-is-Bitmap</a> )	
---------	-----------------------------------	---	--

WebSite	GeeksForGeeks: How to Create Bitmap From View in Android? (	<a href="https://www.geeksforgeeks.org/how-to-create-bitmap-from-view-in-android/">https://www.geeksforgeeks.org/how-to-create-bitmap-from-view-in-android/</a> )
YouTube	tutorialsEU: Glide Tutorial - Working with images in your Android App - Loading and Caching Images (	<a href="https://www.youtube.com/watch?v=_huNIBQ_jLQ">https://www.youtube.com/watch?v=_huNIBQ_jLQ</a> )
YouTube	Stevdza-San: Coil - Modern Image Loading Library   Android Studio Tutorial (	<a href="https://www.youtube.com/watch?v=IBaUjzn2Rgo">https://www.youtube.com/watch?v=IBaUjzn2Rgo</a> )
YouTube	Coding in Flow: How to Load an Image from a Url with Picasso - Android Studio Tutorial (	<a href="https://www.youtube.com/watch?v=Tdb_WSEEZbQ">https://www.youtube.com/watch?v=Tdb_WSEEZbQ</a> )
WebSite	Coil - Official page (	<a href="https://coil-kt.github.io/coil/">https://coil-kt.github.io/coil/</a> )
WebSite	Glide v4 - Official page (	<a href="https://bumptech.github.io/glide/">https://bumptech.github.io/glide/</a> )
WebSite	Picasso - Official page (	<a href="https://square.github.io/picasso/">https://square.github.io/picasso/</a> )

## Level 2

### Android - Permissions:

The app's permissions help support user privacy by protecting restricted data such as system statuses and the user's contact data, there are also restricted actions such as connecting to a paired device and recording audio or using a camera.

Understanding what permissions are

Learning about the different types of permissions

Special permissions

Best practices

### Contents

WebSite	Android Documentation: Permissions on Android (	<a href="https://developer.android.com/guide/topics/permissions/overview">https://developer.android.com/guide/topics/permissions/overview</a> )
WebSite	Android Documentation: - Android Manifest (	<a href="https://developer.android.com/guide/topics/manifest/uses-permission-element">https://developer.android.com/guide/topics/manifest/uses-permission-element</a> )
WebSite	Android Documentation: Request runtime permissions (	<a href="https://developer.android.com/training/permissions/requesting">https://developer.android.com/training/permissions/requesting</a> )
Article	Understanding App Permissions (	<a href="https://guides.codepath.com/android/Understanding-App-Permissions">https://guides.codepath.com/android/Understanding-App-Permissions</a> )
YouTube	Android Developers: Request a permission in Android (	<a href="https://www.youtube.com/watch?v=x38dYUm7tCY">https://www.youtube.com/watch?v=x38dYUm7tCY</a> )
YouTube	Philipp Lackner: PERMISSIONS - Android Fundamentals (	<a href="https://www.youtube.com/watch?v=S4jkcRhembY">https://www.youtube.com/watch?v=S4jkcRhembY</a> )

### Kotlin - Asynchronous:

In asynchronous programming, the functions are not executed in order. We can interrupt the code to get some other information needed to continue execution. This means that the code waits for



another part of the code, and while it waits, it can execute the other parts.

Learning the possibilities to run code asynchronously on Android

Getting to know the java.concurrent package and its solutions

Using Coroutines as a solution for asynchronous code in Kotlin

Understanding and use reactive frameworks like LiveData, Flow, StateFlow, RX, etc

## Contents

WebSite	Android Documentation: Kotlin coroutines on Android (	<a href="https://developer.android.com/kotlin/coroutines">https://developer.android.com/kotlin/coroutines</a> )
WebSite	Android Documentation: Kotlin flows on Android (	<a href="https://developer.android.com/kotlin/flow">https://developer.android.com/kotlin/flow</a> )
WebSite	Android Documentation: StateFlow and SharedFlow (	<a href="https://developer.android.com/kotlin/flow/stateflow-and-sharedflow">https://developer.android.com/kotlin/flow/stateflow-and-sharedflow</a> )
WebSite	Android Documentation: LiveData overview (	<a href="https://developer.android.com/topic/libraries/architecture/livedata">https://developer.android.com/topic/libraries/architecture/livedata</a> )
Article	Fundamentals of RxJava with Kotlin for absolute beginners (	<a href="https://medium.com/@gabriel demattos leon/fundamentals-of-rxjava-with-kotlin-for-absolute-beginners-3d811350b701">https://medium.com/@gabriel demattos leon/fundamentals-of-rxjava-with-kotlin-for-absolute-beginners-3d811350b701</a> )
YouTube	Philipp Lackner: Kotlin Coroutines (	<a href="https://www.youtube.com/playlist?list=PLQkwcJG4YTCQcFEPuYGuv54nYai_lwil">https://www.youtube.com/playlist?list=PLQkwcJG4YTCQcFEPuYGuv54nYai_lwil</a> )
YouTube	Stevdza-San: Kotlin Coroutines (	<a href="https://www.youtube.com/playlist?list=PLSrm9z4zp4mE-o3sPq-PqzGHoFAlsQFI6">https://www.youtube.com/playlist?list=PLSrm9z4zp4mE-o3sPq-PqzGHoFAlsQFI6</a> )
YouTube	CodandoTV(Rods): Configure Networking in your application Retrofit+Okhttp+ Interceptors+Coroutines - FULL VERSION (	<a href="https://youtu.be/D-wckaYH5Do">https://youtu.be/D-wckaYH5Do</a> )

## Kotlin - Communication with APIs:

An API is an interface that software developers use to programmatically interact with software components or resources outside of their own code. An even simpler definition is that an API is the part of a software component that is accessible to other components.

Performing HTTP requests to communicate with online services such as REST API is fundamental to most Android apps. As such, it is important to know the main tools and techniques needed for this type of functionality.

Knowing the most famous libraries in the Android world for making requests - Retrofit, Ktor or Volley

Setting up requests to execute asynchronously

Converting objects to JSON and vice-versa

## Contents

WebSite	Android Documentation: Connect to the network (	<a href="https://developer.android.com/training/basics/network-ops/connecting">https://developer.android.com/training/basics/network-ops/connecting</a> )
Article	Android Volley in 5 mins (	<a href="https://medium.com/@myofficework000/android-volley-in-5-mins-c6c8a62f6796">https://medium.com/@myofficework000/android-volley-in-5-mins-c6c8a62f6796</a> )
Article	Connect android app to the network using HttpURLConnection with Kotlin (	<a href="https://medium.com/@ajay.dewari/connect-android-app-to-the-network-using-httpurlconnection-79a55c8c624e">https://medium.com/@ajay.dewari/connect-android-app-to-the-network-using-httpurlconnection-79a55c8c624e</a> )

Kotlin and Retrofit 2: ArticleTutorial with working codes (	<a href="https://medium.com/mobile-app-development-publication/kotlin-and-retrofit-2-tutorial-with-working-codes-333a4422a890">https://medium.com/mobile-app-development-publication/kotlin-and-retrofit-2-tutorial-with-working-codes-333a4422a890</a>	)
YouTube Coder: How to make API calls using Android Volley Library   Kotlin (	<a href="https://www.youtube.com/watch?v=pfqc7ajaLNE">https://www.youtube.com/watch?v=pfqc7ajaLNE</a>	)
YouTube Philipp Lackner: The Ultimate Retrofit Crash Course (	<a href="https://www.youtube.com/watch?v=t6Sql3WMAk">https://www.youtube.com/watch?v=t6Sql3WMAk</a>	)

## Android - System resources:

The smartphones that are under the Android platform, for the most part, have several system-specific resources, such as cameras, sensors, gps, and others.

Learning what these features are and how to use them

### Contents

WebSite Android Documentation: CameraX overview - Library for using the camera on Android (	<a href="https://developer.android.com/training/camerax">https://developer.android.com/training/camerax</a>	)
WebSite Android Documentation: Sensors - Motion, Positing and Environment sensors (	<a href="https://developer.android.com/guide/topics/sensors">https://developer.android.com/guide/topics/sensors</a>	)
WebSite Android Documentation: Connectivity (Bluetooth, Wi-Fi, Network operations) (	<a href="https://developer.android.com/guide/topics/connectivity">https://developer.android.com/guide/topics/connectivity</a>	)
YouTube Codeible: Implementing Google Maps on Android (	<a href="https://www.youtube.com/watch?v=WouAQmqJl_I">https://www.youtube.com/watch?v=WouAQmqJl_I</a>	)
YouTube tutorialsEU: Camera Android Studio - How to Take Pictures And Use Them in Your Project (	<a href="https://www.youtube.com/watch?v=xZZQ5q5pOp0">https://www.youtube.com/watch?v=xZZQ5q5pOp0</a>	)
YouTube Coder: How to open Camera in Android App (	<a href="https://www.youtube.com/watch?v=xZZQ5q5pOp0">https://www.youtube.com/watch?v=xZZQ5q5pOp0</a>	)

## Kotlin - Dependency Injection:

Dependency Injection is a design pattern in which a class requests dependencies from external sources instead of creating them.

When writing code in Android projects, it is often the case that a feature uses library code, such as Room to save data or Retrofit to make requests to the REST API. These libraries are known as dependencies in our codes, precisely because of their necessity to perform the expected action. Using these libraries easily anywhere in the app can have its challenges, such as providing unique instances and performing all the necessary configuration for correct operation. For this, we use dependency injection tools that make our job easier.

Learning how to use the dependency injection technique and one of the most common Android tools for this - Hilt, Dagger or Koin

### Contents

WebSite Android Documentation: Dependency injection in Android (	<a href="https://developer.android.com/training/dependency-injection?hl=pt-br">https://developer.android.com/training/dependency-injection?hl=pt-br</a>	)
--	---	---

Article	Dependency Injection with Koin (	<a href="https://medium.com/swlh/dependency-injection-with-koin-6b6364dc8dba">https://medium.com/swlh/dependency-injection-with-koin-6b6364dc8dba</a>	)
YouTube	CodingWithMitch: Hilt Dependency Injection (Kotlin Beginner Example) (	<a href="https://www.youtube.com/watch?v=IdelEXu2-2w">https://www.youtube.com/watch?v=IdelEXu2-2w</a>	)
YouTube	Philipp Lackner: The Ultimate Guide to Koin (Dependency Injection with Kotlin) (	<a href="https://www.youtube.com/watch?v=EathumJIWh8">https://www.youtube.com/watch?v=EathumJIWh8</a>	)

## Android - Testing:

Software testing is the process of evaluating and verifying that a software product or application does what it is supposed to do. The benefits of testing include preventing bugs, reducing development costs and improving performance.

Using unit tests

Using integration testing

Using instrumented testing

Using mocks to facilitate test implementation with dependencies

Implementing tests with Dependency Injection tools such as Hilt

### Contents

	Android Documentation:		
WebSite	Fundamentals of testing	<a href="https://developer.android.com/training/testing/fundamentals">https://developer.android.com/training/testing/fundamentals</a>	)
	Android apps (		
WebSite	Android Documentation: What to test in Android (	<a href="https://developer.android.com/training/testing/fundamentals/what-to-test">https://developer.android.com/training/testing/fundamentals/what-to-test</a>	)
WebSite	Android Documentation: Testing your Compose layout (	<a href="https://developer.android.com/jetpack/compose/testing">https://developer.android.com/jetpack/compose/testing</a>	)
WebSite	Android Documentation: Build local unit tests (	<a href="https://developer.android.com/training/testing/local-tests">https://developer.android.com/training/testing/local-tests</a>	)
WebSite	Android Documentation: Build instrumented tests (	<a href="https://developer.android.com/training/testing/instrumented-tests">https://developer.android.com/training/testing/instrumented-tests</a>	)
Article	The basics of Unit and Instrumentation Testing on Android (	<a href="https://medium.com/@ali.muzaffar/the-basics-of-unit-and-instrumentation-testing-on-android-7f3790e77bd">https://medium.com/@ali.muzaffar/the-basics-of-unit-and-instrumentation-testing-on-android-7f3790e77bd</a>	)
YouTube	Philipp Lackner: Testing on Android (	<a href="https://www.youtube.com/playlist?list=PLQkwcJG4YTCsYJ13G4kVIJ10X5zisB2Lq">https://www.youtube.com/playlist?list=PLQkwcJG4YTCsYJ13G4kVIJ10X5zisB2Lq</a>	)
YouTube	Cheezy Code: Android Unit Testing Playlist - JUnit, Espresso, Mockito with Kotlin (	<a href="https://www.youtube.com/playlist?list=PLQkwcJG4YTCsYJ13G4kVIJ10X5zisB2Lq">https://www.youtube.com/playlist?list=PLQkwcJG4YTCsYJ13G4kVIJ10X5zisB2Lq</a>	)

## Level 3

### Android - Architecture:

The Android architecture guide covers practices and recommended architecture for building robust apps with high production quality. Organizing your code base into contained and flexibly coupled

parts (Modularization).

Learning what the MVVM (Model-View-ViewModel) architecture is

UI, data and domain layers

Data flow within the app

How to use the ViewModel

Architecture recommendations, repository pattern, offline-first

State management

Dependency injection

Improving app user experience with ViewModel

Learning what modularization is

Benefits of modularization

## Contents

WebSite Android Documentation: Guide to app architecture ( <https://developer.android.com/topic/architecture>)

Android

WebSite Documentation: <https://developer.android.com/topic/libraries/architecture/viewmodel>  
ViewModel overview (

WebSite Android Documentation: UI layer (<https://developer.android.com/topic/architecture/ui-layer>)

WebSite Android Documentation: Domain layer ( <https://developer.android.com/topic/architecture/domain-layer>)

WebSite Android Documentation: Data layer (<https://developer.android.com/topic/architecture/data-layer>)

WebSite Android Documentation: Build an offline-first app ( <https://developer.android.com/topic/architecture/data-layer/offline-first>)

WebSite Android Documentation: Guide to Android app modularization ( <https://developer.android.com/topic/modularization>)

Android Documentation:

WebSite Recommendations for Android architecture ( <https://developer.android.com/topic/architecture/recommendations>)

YouTube Philipp Lackner: MVVM in 100 Seconds (<https://www.youtube.com/watch?v=-xTqfilaYow>)

YouTube Indently: The ViewModel & LiveData Tutorial in Android Studio (Kotlin) ( <https://www.youtube.com/watch?v=N7J27pBTtUl>)

YouTube Stevdza-San: LiveData Explained - Android Architecture Component ( <https://www.youtube.com/watch?v=suC0OM5gGAA>)

YouTube Stevdza-San: ViewModel Explained - Android Architecture Component ( <https://www.youtube.com/watch?v=orH4K6qBzvE>)

## Android - Core App quality:

When writing an Android App, there are a number of requirements to ensure an expected quality, such as compatibility of the App with the navigation expected by the Android system, or the

gestures, etc.

The requirements are classified as - Visual experience, Functionality, Performance and stability, Privacy and security, Google Play, and Testing procedure

Fulfilling all or almost all requirements means that the App has a higher quality for the users

## Contents

WebSite	Android Documentation: Core app quality (	<a href="https://developer.android.com/docs/quality-guidelines/core-app-quality">https://developer.android.com/docs/quality-guidelines/core-app-quality</a>	)
WebSite	Android Documentation: Deliver high performing user experiences (	<a href="https://developer.android.com/quality">https://developer.android.com/quality</a> )	
WebSite	Android Documentation: Get started with large screens (	<a href="https://developer.android.com/guide/topics/large-screens/get-started-with-large-screens">https://developer.android.com/guide/topics/large-screens/get-started-with-large-screens</a>	)
WebSite	Android Documentation: Build for billions (	<a href="https://developer.android.com/docs/quality-guidelines/build-for-billions">https://developer.android.com/docs/quality-guidelines/build-for-billions</a>	)
YouTube	Android Developers: Introduction to Android app quality (	<a href="https://www.youtube.com/watch?v=QxyvS1oZOUs">https://www.youtube.com/watch?v=QxyvS1oZOUs</a>	)
YouTube	Android Developers: Android app quality - Visual experience (	<a href="https://www.youtube.com/watch?v=uUWqUp_HeNs">https://www.youtube.com/watch?v=uUWqUp_HeNs</a>	)

## Android - CI/CD:

CI/CD is short for Continuous Integration/Continuous Delivery. It is a software development practice that aims to make code integration more efficient through automated builds and testing.

When implementing new App functionality, we need to ensure that all deliveries will work correctly. To do this, we can use continuous integration and continuous delivery techniques, this way we speed up the evolution of the App and try to ensure the expected behaviors at the same time.

Knowing one of the tools to do continuous delivery, such as Firebase Test Lab, Jenkins, GitHub Actions, etc

## Contents

WebSite	Firebase: Start testing with continuous integration (CI) systems (	<a href="https://firebase.google.com/docs/test-lab/android/continuous">https://firebase.google.com/docs/test-lab/android/continuous</a>	)
Article	The Anatomy of Android Continuous Delivery (	<a href="https://medium.com/the-telegraph-engineering/android-continuous-delivery-fb41da63176">https://medium.com/the-telegraph-engineering/android-continuous-delivery-fb41da63176</a>	)
WebSite	CodePath: Automating Publishing to the Play Store (	<a href="https://guides.codepath.com/android/automating-publishing-to-the-play-store">https://guides.codepath.com/android/automating-publishing-to-the-play-store</a>	)
YouTube	Philipp Lackner: Learn to Automate Tasks for Android with CI/CD (	<a href="https://www.youtube.com/watch?v=QLsgkxH-O2I">https://www.youtube.com/watch?v=QLsgkxH-O2I</a>	)
YouTube	Android Developer Tips: What is the CI/CD Pipeline from an Android app perspective? — Part 1: CI (	<a href="https://www.youtube.com/watch?v=4nVRJ9uIKJQ">https://www.youtube.com/watch?v=4nVRJ9uIKJQ</a>	)
YouTube	Android Developer Tips: What is the CI/CD Pipeline from an Android app perspective? — Part 2: CD (	<a href="https://www.youtube.com/watch?v=bYi-nl1lfc0">https://www.youtube.com/watch?v=bYi-nl1lfc0</a>	)

## Android - App optimization:

When generating an App, there are some optimization details to make it fast, smaller and optimized, such as removing unnecessary code, obfuscation that reduces code names, and optimization that applies more aggressive strategies to further reduce the app.

Learning how to enable each optimization in the Android plugin for Gradle

Using proguard to perform a more aggressive optimization

Activating multidex so that the app will be able to get more than 64000 methods and be able to use the optimization techniques

Knowing and using helper tools to increase the performance of the App

### Contents

WebSite	Android Documentation: Shrink, obfuscate, and optimize your app (	<a href="https://developer.android.com/studio/build/shrink-code">https://developer.android.com/studio/build/shrink-code</a> )
WebSite	Android Documentation: Enable multidex for apps with over 64K methods (	<a href="https://developer.android.com/studio/build/multidex">https://developer.android.com/studio/build/multidex</a> )
WebSite	Android Documentation: Guide to app performance (	<a href="https://developer.android.com/topic/performance">https://developer.android.com/topic/performance</a> )
YouTube	Philipp Lackner: Shrink, Optimize and Secure Your App With R8 & ProGuard - Full Guide (	<a href="https://www.youtube.com/watch?v=bgpyuuzMlo0">https://www.youtube.com/watch?v=bgpyuuzMlo0</a> )
YouTube	Philipp Lackner: 5 Ways to Boost Your Android App's Performance (	<a href="https://www.youtube.com/watch?v=epkAPnF5qrk">https://www.youtube.com/watch?v=epkAPnF5qrk</a> )

## Java - Fundamentals:

Java is a widely-used programming language for coding web applications. Java is a multi-platform, object-oriented, and network-centric language that can be used as a platform in itself. It is a fast, secure, reliable programming language for coding everything from mobile apps and enterprise software to big data applications and server-side technologies.

Knowing the primitive types

Declaring variables, considering the different types

Using conditional structures ('if', 'else')

Knowing the assignment and comparison operators

Using repetition structures and loops ('while', 'for')

Using functions, passing parameters and arguments

Manipulating methods

Manipulating arrays and lists

Getting data from an API

Making asynchronous 'Future' calls, etc

Creating constructors

### Contents

WebSiteJava - Documentation (<https://dev.java/learn/>)

WebSiteWhat Is Java? (<https://aws.amazon.com/what-is/java/>)

WebSiteJava - Documentation - Primitive types (<https://dev.java/learn/creating-primitive-type-variables-in-your-programs/>)

WebSiteJava - Documentation - Creating Arrays (<https://dev.java/learn/creating-arrays-in-your-programs/>)

WebSiteJava - Documentation - Creating Classes (<https://dev.java/learn/creating-classes/>)

WebSiteJava - Documentation - Defining Methods (<https://dev.java/learn/defining-methods/>)

WebSiteJava - Documentation - Creating and Using Objects (<https://dev.java/learn/creating-and-using-objects/>)

WebSiteW3Schools: Java If ... Else ([https://www.w3schools.com/java/java\\_conditions.asp](https://www.w3schools.com/java/java_conditions.asp))

WebSiteW3Schools: Java While Loop ([https://www.w3schools.com/java/java\\_while\\_loop.asp](https://www.w3schools.com/java/java_while_loop.asp))

WebSiteW3Schools: Java For Loop ([https://www.w3schools.com/java/java\\_for\\_loop.asp](https://www.w3schools.com/java/java_for_loop.asp))

ArticleAsync in Java (<https://medium.com/cognizantsoftvision-guildhall/async-in-java-80a7240fefa8>)

YouTubeAlex Lee: Learn Java in 14 Minutes (<https://www.youtube.com/watch?v=RRubcjpTkks>)

YouTubeProgramming with Mosh: Java Tutorial for Beginners (<https://www.youtube.com/watch?v=elrMbAQSU34>)

YouTubeBro Code: Java Full Course (FREE) ([https://www.youtube.com/watch?v=xk4\\_1vDrzzo](https://www.youtube.com/watch?v=xk4_1vDrzzo))

## Android - Debugging:

During the development of an app, it is quite common to have unexpected bugs, as well as a lack of understanding of the reason for the bug. To speed up the analysis and investigation of problems or unexpected behavior in the app, we need to learn how to debug an Android app.

Learning how to debug an app in Android Studio

Learning how to enable debugging on physical devices

Using logs to identify events

Analyzing stack tracks

Inspecting layout, system resources like processor, memory, and network

Debugging the app database and pre-compiled APK files

Analyzing the build with APK parser

## Contents

WebSiteAndroid Documentation: Debug your app (<https://developer.android.com/studio/debug>)

WebSiteAndroid Documentation: Configure on-device developer options (<https://developer.android.com/studio/debug/dev-options>)

WebSiteAndroid Documentation: View logs with Logcat (<https://developer.android.com/studio/debug/logcat>)

WebSiteAndroid Documentation: Analyze a stack trace (<https://developer.android.com/studio/debug/stacktraces>)



Android Documentation: Debug your WebSite	layout with Layout Inspector and Layout Validation (	<a href="https://developer.android.com/studio/debug/layout-inspector">https://developer.android.com/studio/debug/layout-inspector</a> )
Android Documentation: Inspeccionar o WebSite	tráfego de rede com o Network Inspector (	<a href="https://developer.android.com/studio/debug/network-profiler?hl=pt-br">https://developer.android.com/studio/debug/network-profiler?hl=pt-br</a> )
Android Documentation: Inspect WebSite	network traffic with the Network Inspector (	<a href="https://developer.android.com/studio/debug/network-profiler">https://developer.android.com/studio/debug/network-profiler</a> )
Android Documentation: Debug pre-built WebSite	APKs (	<a href="https://developer.android.com/studio/debug/apk-debugger">https://developer.android.com/studio/debug/apk-debugger</a> )
Android Documentation: Capture and read WebSite	bug reports (	<a href="https://developer.android.com/studio/debug/bug-report">https://developer.android.com/studio/debug/bug-report</a> )
Android Documentation: Analyze your WebSite	build with the APK Analyzer (	<a href="https://developer.android.com/studio/debug/apk-analyzer">https://developer.android.com/studio/debug/apk-analyzer</a> )
Philipp Lackner: The Full Guide to Debugging Your YouTube	Android Apps (	<a href="https://www.youtube.com/watch?v=ln5hc-zprEM">https://www.youtube.com/watch?v=ln5hc-zprEM</a> )

## Android - Security and Monitoring:

As your project grows, it becomes more visible, attracting both a larger audience and potential attackers. Application security, from the earliest stages of development, is crucial to protecting the integrity of the project and the privacy of users. Adopting robust security practices and keeping up to date is essential to avoid threats and guarantee a safe experience for your company and its users

Making sure you understand your project's needs

Monitoring is fundamental to understanding how your application operates in production.

Understanding the user flow and identifying possible funnels helps to improve your project strategy and can be a determining factor in the application's success.

The ability to anticipate and predict problem scenarios before they affect the end user is a crucial factor.

### Contents

Documentação: Security WebSite	guidelines (	<a href="https://developer.android.com/privacy-and-security/security-tips">https://developer.android.com/privacy-and-security/security-tips</a> )
Documentação: Run the WebSite	monitor app (	<a href="https://developer.android.com/games/sdk/performance-tuner/unity/run-monitor-app">https://developer.android.com/games/sdk/performance-tuner/unity/run-monitor-app</a> )
Github(Rods) - Mobile developer WebSite	security roadmap (	<a href="https://www.youtube.com/watch?v=bgpyuuzMlo0&amp;ab_channel=PhilippLackner">https://www.youtube.com/watch?v=bgpyuuzMlo0&amp;ab_channel=PhilippLackner</a> )
Philipp Lackner: Shrink, Optimize and Secure YouTube	Your App With R8 & ProGuard - Full Guide (	<a href="https://www.youtube.com/watch?v=bgpyuuzMlo0&amp;ab_channel=PhilippLackner">https://www.youtube.com/watch?v=bgpyuuzMlo0&amp;ab_channel=PhilippLackner</a> )
Philipp Lackner: THIS Is How You Measure the YouTube	Performance of Your Android App (	<a href="https://www.youtube.com/watch?v=XHz_cFwdfom&amp;ab_channel=PhilippLackner">https://www.youtube.com/watch?v=XHz_cFwdfom&amp;ab_channel=PhilippLackner</a> )
Slides about Android WebSite	Development - by Moro (	<a href="https://github.com/gabrielbmoro/slides-about-android-development/blob/main/README.md#security-">https://github.com/gabrielbmoro/slides-about-android-development/blob/main/README.md#security-</a> )



## Auxiliary Skill: Infrastructure and good practices

### Git & GitHub - Fundamentals:

Git is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency.

GitHub is a hosting service for software development and version control using Git.

Creating a repository

Cloning a repository

Committing, pushing and pulling to and from the repository

Reversing a commit

Creating branches and pull requests

Handling merge and conflicts

### Contents

WebSiteGit Reference Book (<https://git-scm.com/book/en/v2>)

WebSiteGitHub Documentation (<https://docs.github.com/en>)

WebSiteGitHub Pages Documentation ( <https://docs.github.com/en/pages/getting-started-with-github-pages/about-github-pages> )

WebSiteW3Schools: Git Tutorial (<https://www.w3schools.com/git/default.asp?remote=github>)

WebSiteGit School - Visualizing Git (<https://git-school.github.io/visualizing-git/>)

WebSiteDangit, Git!?! (<https://dangitgit.com/>)

ArticleGit Tutorial - Explore The Commands And Operations In Git ( <https://medium.com/edureka/git-tutorial-da652b566ece> )

ArticleGit and Github Quickstart Tutorial ( <https://medium.com/@prashantramnyc/git-and-github-quickstart-tutorial-654a71594dca> )

ArticleGetting Started with Git and GitHub: A Complete Tutorial for Beginner ( <https://towardsdatascience.com/learn-basic-git-commands-for-your-data-science-works-2a75396d530d> )

YouTubeProgramming with Mosh: Git Tutorial for Beginners - Learn Git in 1 Hour ( <https://www.youtube.com/watch?v=8JJ101D3knE> )

YouTubefreeCodeCamp.org: Git and GitHub for Beginners - Crash Course ( <https://www.youtube.com/watch?v=RGQj5yH7evk> )

YouTubeKevin Stratvert: Git and GitHub for Beginners Tutorial ( <https://www.youtube.com/watch?v=tRZGeaHPoaw> )

YouTubeTech With Tim: Git Tutorial for Beginners - Git & GitHub Fundamentals In Depth ( <https://www.youtube.com/watch?v=DVRQoVRzMIY> )

### HTTP - Fundamentals:

HTTP stands for Hyper Text Transfer Protocol. Communication between client computers and web servers is done by sending HTTP Requests and receiving HTTP Responses.

Understanding the difference between HTTP verbs

Testing requests and checking the status codes in the browser

Learning how to make a HTTP request on the command line with WGET

Downloading an image with WGET

Performing a POST

## Contents

WebSiteW3Schools: What is HTTP? ([https://www.w3schools.com/whatis/whatis\\_http.asp](https://www.w3schools.com/whatis/whatis_http.asp))

WebSite MDN Web Docs: An overview of HTTP (<https://developer.mozilla.org/en-US/docs/Web/HTTP/Overview>)

WebSite MDN Web Docs: HTTP request methods (<https://developer.mozilla.org/en-US/docs/Web/HTTP/Methods>)

WebSiteHTTP Cats (<https://http.cat/>)

WebSiteHTTP Dogs (<https://http.dog/>)

Article HTTP codes as Valentine's Day comics (<https://medium.com/@hanilim/http-codes-as-valentines-day-comics-8c03c805faa0>)

Article Here Are the most popular ways to make an HTTP request in JavaScript (<https://medium.com/free-code-camp/here-is-the-most-popular-ways-to-make-an-http-request-in-javascript-954ce8c95aaa>)

YouTube Traversy Media: HTTP Crash Course & Exploration (<https://www.youtube.com/watch?v=iYM2zFP3Zn0>)

YouTube Curious Code: HTTP Request Methods - GET, POST, PUT, DELETE (<https://www.youtube.com/watch?v=tkfVQK6UxDI>)

YouTube freeCodeCamp.org: Postman Beginner's Course - API Testing (<https://www.youtube.com/watch?v=VywxIQ2ZXw4>)

## JSON:

JSON stands for JavaScript Object Notation. It is a text format for storing and transporting data.

Creating an object

Transforming an object into a string

Transforming a string into an object

Manipulating an object

## Contents

WebSiteW3Schools: JSON ([https://www.w3schools.com/js/js\\_json\\_intro.asp](https://www.w3schools.com/js/js_json_intro.asp))

WebSite MDN Web Docs: JSON ([https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global\\_Objects/JSON](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/JSON))

WebSite MDN Web Docs: Working with JSON (<https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Objects/JSON>)

YouTube Web Dev Simplified: Learn JSON in 10 Minutes (<https://www.youtube.com/watch?v=iiADhChRriM>)

YouTubeTraversy Media: JSON Crash Course (<https://www.youtube.com/watch?v=w1CWzNtE-M>)

## Design Patterns:

In software engineering, a Design Pattern is a general, reusable solution to a commonly occurring problem within a given context in software design. It is a description or template for how to solve a problem that can be used in many different situations. Design Patterns are formalized best practices that the programmer can use to solve common problems when designing an application or system.

Getting familiarized with and applying the main Design Patterns

### Contents

WebSiteWhat's a design pattern? (<https://refactoring.guru/design-patterns/what-is-pattern>)

Article Design Patterns — Introduction ( <https://henriquesd.medium.com/design-patterns-introduction-220f811db857> )

Article What Are Design Patterns and Do I Need Them? ( <https://www.developer.com/design/what-are-design-patterns-and-do-i-need-them/> )

Article The 3 Types of Design Patterns All Developers Should Know (with code examples of each) ( <https://www.freecodecamp.org/news/the-basic-design-patterns-all-developers-need-to-know/> )

Article 10 Design Patterns every Software Architect and Software Engineer must know ( <https://ravindraelicherla.medium.com/10-design-patterns-every-software-architect-must-know-b33237bc01c2> )

Article Modern-Day Architecture Design Patterns for Software Professionals ( <https://betterprogramming.pub/modern-day-architecture-design-patterns-for-software-professionals-9056ee1ed977> )

YouTube Web Dev Simplified: Design Patterns (Playlist) ( [https://www.youtube.com/playlist?list=PLZIA0Gpn\\_vH\\_CthENcPCM0Dww6a5XYC7f](https://www.youtube.com/playlist?list=PLZIA0Gpn_vH_CthENcPCM0Dww6a5XYC7f) )

YouTube Programming with Mosh: Design Patterns in Plain English ( [https://www.youtube.com/watch?v=NU\\_1StN5Tkk](https://www.youtube.com/watch?v=NU_1StN5Tkk) )

YouTube Fireship: 10 Design Patterns Explained in 10 Minutes ( [https://www.youtube.com/watch?v=tv-\\_1er1mWl](https://www.youtube.com/watch?v=tv-_1er1mWl) )

YouTube Traversy Media: 5 Design Patterns Every Engineer Should Know ( <https://www.YOUTUBE.com/watch?v=AWOf6Wo6gtg> )

## Command Line - Fundamentals:

CLI is a command line program that accepts text input to execute operating system functions.

Knowing the most important commands

### Contents

WebSite W3Schools: What is Command Line Interface (CLI)? ( [https://www.w3schools.com/whatis/whatis\\_cli.asp](https://www.w3schools.com/whatis/whatis_cli.asp))

WebSite Microsoft Docs: Using command line arguments for Windows Terminal ( <https://docs.microsoft.com/en-us/windows/terminal/command-line-arguments?tabs=windows> )

Article	Advanced CLI: Commands You Should Know as a Developer (	<a href="https://betterprogramming.pub/advanced-cli-commands-you-should-know-as-a-developer-7bc48c752a5e">https://betterprogramming.pub/advanced-cli-commands-you-should-know-as-a-developer-7bc48c752a5e</a>	)
YouTube	freeCodeCamp.org: Command Line Crash Course (	<a href="https://www.youtube.com/watch?v=yz7nYInXLfE">https://www.youtube.com/watch?v=yz7nYInXLfE</a>	)
YouTube	Traversy Media: Command Line Crash Course For Beginners - Terminal Commands (	<a href="https://www.youtube.com/watch?v=uwAqEzhyjtw">https://www.youtube.com/watch?v=uwAqEzhyjtw</a>	)

## Cloud - Fundamentals:

Cloud, or cloud computing, is the distribution of computing services over the Internet using a pay-as-you-go pricing model. A cloud is composed of various computing resources, ranging from the computers themselves (or instances, in cloud terminology) to networks, storage, databases, and everything around them. In other words, everything that is normally needed to set up the equivalent of a server room, or even a complete data center, will be ready to use, configured, and run.

Knowing the difference between IaaS, PaaS and SaaS

Knowing the largest cloud providers

Specializing in a specific provider of your choice

## Contents

WebSite	Microsoft Azure: What is cloud computing? (	<a href="https://azure.microsoft.com/en-us/resources/cloud-computing-dictionary/what-is-cloud-computing/">https://azure.microsoft.com/en-us/resources/cloud-computing-dictionary/what-is-cloud-computing/</a>	)
WebSite	Amazon AWS: What is cloud computing? (	<a href="https://aws.amazon.com/en/what-is-cloud-computing/">https://aws.amazon.com/en/what-is-cloud-computing/</a>	)
Article	A beginner's guide to the basics of what cloud computing is about (	<a href="https://scientya.com/a-beginners-guide-to-the-basics-of-what-cloud-computing-is-about-e8b3b7f25a30/">https://scientya.com/a-beginners-guide-to-the-basics-of-what-cloud-computing-is-about-e8b3b7f25a30/</a>	)
Article	Cloud Computing for Beginners (	<a href="https://medium.com/hackernoon/cloud-computing-for-beginners-85d168959afb/">https://medium.com/hackernoon/cloud-computing-for-beginners-85d168959afb/</a>	)
Article	What are Cloud Computing Services [IaaS, CaaS, PaaS, FaaS, SaaS] (	<a href="https://medium.com/@nnilesh7756/what-are-cloud-computing-services-iaas-caas-paas-faas-saas-ac0f6022d36e">https://medium.com/@nnilesh7756/what-are-cloud-computing-services-iaas-caas-paas-faas-saas-ac0f6022d36e</a>	)
YouTube	Simplilearn: Cloud Computing Tutorial for Beginners (	<a href="https://www.youtube.com/watch?v=RWgW-CgdIk0">https://www.youtube.com/watch?v=RWgW-CgdIk0</a>	)
YouTube	Amazon Web Services: What is Cloud Computing? - Amazon Web Services (	<a href="https://www.youtube.com/watch?v=mxT233EdY5c">https://www.youtube.com/watch?v=mxT233EdY5c</a>	)
YouTube	Ecourse Review: Cloud Computing Services Models - IaaS PaaS SaaS Explained (	<a href="https://www.youtube.com/watch?v=36zducUX16w/">https://www.youtube.com/watch?v=36zducUX16w/</a>	)

## SOLID:

SOLID has five principles that are considered best practices in software development that help programmers write cleaner code by separating responsibilities, reducing coupling, easing refactoring, and encouraging code reuse.

## Contents

Article	The S.O.L.I.D Principles in Pictures (	<a href="https://medium.com/backticks-tildes/the-s-o-l-i-d-principles-in-pictures-b34ce2f1e898">https://medium.com/backticks-tildes/the-s-o-l-i-d-principles-in-pictures-b34ce2f1e898</a>	)
Article	SOLID Principles every Developer Should Know (	<a href="https://blog.bitsrc.io/solid-principles-every-developer-should-know-b3bfa96bb688">https://blog.bitsrc.io/solid-principles-every-developer-should-know-b3bfa96bb688</a>	)
YouTube	Fireship: Solid in 100 Seconds (	<a href="https://www.youtube.com/watch?v=q0BGgQJcp7w">https://www.youtube.com/watch?v=q0BGgQJcp7w</a>	)
YouTube	A Dev' Story: SOLID Design Principles Explained in a Nutshell (	<a href="https://www.youtube.com/watch?v=69sfWNzxTMc">https://www.youtube.com/watch?v=69sfWNzxTMc</a>	)
YouTube	Web Dev Simplified: Single Responsibility Principle Explained - SOLID Design Principles (	<a href="https://www.youtube.com/watch?v=UQqY3_6Epbq">https://www.youtube.com/watch?v=UQqY3_6Epbq</a>	)

## Clean Architecture:

Clean architecture is a way of developing software, such that just by looking at the source code of a program, you should be able to tell what the program does.

### Contents

Article	The Clean Architecture—Beginner's Guide (	<a href="https://betterprogramming.pub/the-clean-architecture-beginners-guide-e4b7058c1165">https://betterprogramming.pub/the-clean-architecture-beginners-guide-e4b7058c1165</a>	)
Article	Thoughts on Clean Architecture (	<a href="https://medium.com/android-news/thoughts-on-clean-architecture-b8449d9d02df">https://medium.com/android-news/thoughts-on-clean-architecture-b8449d9d02df</a>	)
Article	Clean Architecture, the right way (	<a href="https://medium.com/gdg-vit/clean-architecture-the-right-way-d83b81ecac6">https://medium.com/gdg-vit/clean-architecture-the-right-way-d83b81ecac6</a>	)
Article	Why use a Clean Architecture (	<a href="https://www.mytaskpanel.com/the-5-advantages-of-using-a-clean-architecture-all-you-need-to-know/">https://www.mytaskpanel.com/the-5-advantages-of-using-a-clean-architecture-all-you-need-to-know/</a>	)
Article	Clean Architecture – Build Software like an Artisan (	<a href="https://flexiple.com/developers/clean-architecture-build-software-like-an-artisan/">https://flexiple.com/developers/clean-architecture-build-software-like-an-artisan/</a>	)
Article	React Clean Architecture (	<a href="https://kpiteng.medium.com/react-clean-architecture-e4144a0788b6">https://kpiteng.medium.com/react-clean-architecture-e4144a0788b6</a>	)
Article	Clean Architecture with Java 11 (	<a href="https://medium.com/slalom-build/clean-architecture-with-java-11-f78bba431041">https://medium.com/slalom-build/clean-architecture-with-java-11-f78bba431041</a>	)
Article	A detailed guide on developing Android apps using the Clean Architecture pattern (	<a href="https://medium.com/@dmilicic/a-detailed-guide-on-developing-android-apps-using-the-clean-architecture-pattern-d38d71e94029">https://medium.com/@dmilicic/a-detailed-guide-on-developing-android-apps-using-the-clean-architecture-pattern-d38d71e94029</a>	)
YouTube	CodeOpinion: Clean Architecture Example & Breakdown (	<a href="https://www.youtube.com/watch?v=Ys_W6MyWOCw">https://www.youtube.com/watch?v=Ys_W6MyWOCw</a>	)
YouTube	Women Who Code: Clean Architecture Concepts (	<a href="https://www.youtube.com/watch?v=ouBSPdzbzvw">https://www.youtube.com/watch?v=ouBSPdzbzvw</a>	)
YouTube	CodandoTV(Rods) - Simplifying Clean Architecture +MVVM in your mobile application - COMPLETE GUIDE (	<a href="https://youtu.be/8ehIZfyN1S0?si=I7-I5l4zsnYLIJSjD">https://youtu.be/8ehIZfyN1S0?si=I7-I5l4zsnYLIJSjD</a>	)

## Firebase:

Firebase is a Backend-as-a-Service (BaaS) app development platform that provides hosted backend services such as a realtime database, cloud storage, authentication, crash reporting, machine learning, remote configuration, and hosting for your static files.

Understanding how to install Firebase

Getting acquainted with Firebase documentation

Learning about the Firebase tools available

## Contents

WebSiteFlutter: Firebase (<https://docs.flutter.dev/development/data-and-backend/firebase>)

ArticleFirebase — Flutter (<https://medium.com/google-developer-experts/firebase-70fd12309fd1>)

Article Firebase Android Series: Learning <https://proandroiddev.com/firebase-android-series-firebase-from-zero-to-hero-3bacbdf8e048> )

WebSiteFirebase - Case studies (<https://firebase.google.com/use-cases>)

WebSite Add Firebase to your Flutter app ( <https://firebase.google.com/docs/flutter/setup?platform=android> )

WebSiteAdd Firebase to your Android project (<https://firebase.google.com/docs/android/setup>)

YouTube Firebase: Getting started with Firebase on Flutter - Firecasts ( [https://www.youtube.com/watch?v=EXp0gg9kGxI&ab\\_channel=Firebase](https://www.youtube.com/watch?v=EXp0gg9kGxI&ab_channel=Firebase) )

YouTube Firebase: Getting started with Firebase on Android - Firecasts ( [https://www.youtube.com/watch?v=dRYnm\\_k3w1w](https://www.youtube.com/watch?v=dRYnm_k3w1w) )

YouTubeJohannes Milke: Flutter Firebase CRUD ([https://www.youtube.com/watch?v=ErP\\_xomHKTW](https://www.youtube.com/watch?v=ErP_xomHKTW))

YouTube GeeksforGeeks: User Authentication and CRUD Operation with Firebase Realtime Database in Android ( <https://www.youtube.com/watch?v=-Gvpf8tXpbc> )

WebSiteFirebase - Pricing plans (<https://firebase.google.com/pricing>)

## Auxiliary Skill: UX & Design

### Material Design:

Material Design is Google's open source design system, where it gives you components with certain usage patterns and some customization for your apps.

User Experience Foundations

Customizing your components

Adaptive layouts

## Contents

WebSiteMaterial 3 - Get Started (<https://m3.material.io/get-started>)

WebSiteMaterial 3 - Foundations (<https://m3.material.io/foundations>)

WebSiteMaterial 3 - Styles (<https://m3.material.io/styles>)

WebSiteMaterial 3 - Components (<https://m3.material.io/components>)

Article Introduction to Material Design in Android ( <https://www.geeksforgeeks.org/introduction-to-material-design-in-android/> )

Content Philipp Lackner: Introduction to Material 3 (Color Theming, Typography, Shapes) ( <https://www.youtube.com/watch?v=I3eT32LXAKc> )

## Design Systems:

A design system is a collection of reusable components, guided by clear standards, that can be assembled together to build applications.

Creating and maintaining libraries that will be consumed and used as a standard for building a project

### Contents

Article	Everything you need to know about Design Systems (	<a href="https://uxdesign.cc/everything-you-need-to-know-about-design-systems-54b109851969">https://uxdesign.cc/everything-you-need-to-know-about-design-systems-54b109851969</a>	)
Article	A comprehensive guide to design systems (	<a href="https://www.invisionapp.com/inside-design/guide-to-design-systems/">https://www.invisionapp.com/inside-design/guide-to-design-systems/</a>	)
Article	What is a UX Design System? (	<a href="https://fuzzymath.com/blog/what-is-ux-design-system/">https://fuzzymath.com/blog/what-is-ux-design-system/</a>	)
Article	Creating a design system in Figma — Tutorial 1 (	<a href="https://medium.com/@johan_ronsse/creating-a-design-system-in-figma-the-not-so-definite-guide-tutorial-1-8aa6801101cb">https://medium.com/@johan_ronsse/creating-a-design-system-in-figma-the-not-so-definite-guide-tutorial-1-8aa6801101cb</a>	)
YouTube	DesignerUp: 5 Best Design Systems and How to Learn (and Steal) From Them (	<a href="https://www.youtube.com/watch?v=BISC15OPeGA">https://www.youtube.com/watch?v=BISC15OPeGA</a>	)
YouTube	DesignCourse: What is a Design System? Design Systems 101 for Designers (	<a href="https://www.youtube.com/watch?v=wc5krC28ynQ">https://www.youtube.com/watch?v=wc5krC28ynQ</a>	)
YouTube	vaexperience: Design Systems - What are They and How to Get Started (	<a href="https://www.youtube.com/watch?v=I6YuGE6EjJA">https://www.youtube.com/watch?v=I6YuGE6EjJA</a>	)

## Color systems:

Defining a color palette that makes sense for a given interface

### Contents

Article	Color Theory: There is more than one set of universal Primary Colors (	<a href="https://medium.com/upskilling/color-theory-there-is-more-than-one-set-of-universal-primary-colors-debunking-the-myths-28140a7866c9">https://medium.com/upskilling/color-theory-there-is-more-than-one-set-of-universal-primary-colors-debunking-the-myths-28140a7866c9</a>	)
Article	The Science of Color (	<a href="https://medium.com/100-days-of-product-design/the-science-of-color-bd0f057c08ea">https://medium.com/100-days-of-product-design/the-science-of-color-bd0f057c08ea</a>	)
Article	RGB vs. CMYK: A guide to color systems for designers (	<a href="https://medium.com/envato/rgb-vs-cmyk-a-guide-to-color-systems-for-designers-6be8c1ed8554">https://medium.com/envato/rgb-vs-cmyk-a-guide-to-color-systems-for-designers-6be8c1ed8554</a>	)
Article	Color in UI Design: A (Practical) Framework (	<a href="https://medium.com/@erikdkennedy/color-in-ui-design-a-practical-framework-e18cacd97f9e">https://medium.com/@erikdkennedy/color-in-ui-design-a-practical-framework-e18cacd97f9e</a>	)
Article	Basic UI color guide (	<a href="https://blog.prototypr.io/basic-ui-color-guide-7612075cc71a">https://blog.prototypr.io/basic-ui-color-guide-7612075cc71a</a>	)
YouTube	Rachel How: How to pick the right colors for your website or app (	<a href="https://www.youtube.com/watch?v=ewRYw4pnKQU">https://www.youtube.com/watch?v=ewRYw4pnKQU</a>	)
YouTube	Flux Academy: How to Choose Colors (Easy 3-Step Process) (	<a href="https://www.youtube.com/watch?v=KMS3VwGh3HY">https://www.youtube.com/watch?v=KMS3VwGh3HY</a>	)
YouTube	Jesse Showalter: 60-30-10 Color Rule (	<a href="https://www.youtube.com/watch?v=UWwNIMHFdW4">https://www.youtube.com/watch?v=UWwNIMHFdW4</a>	)



DesignerUp: Super Practical Guide to Color Theory, Color Models and Perfect Color Palettes ( <https://www.youtube.com/watch?v=GyVMoejbGFg> )

## How to use Fonts:

Choosing the most appropriate font for a given project

### Contents

Article Top 10 UI Fonts for Web & Mobile ( <https://blog.prototypr.io/top-10-ui-fonts-for-web-mobile-a8488e561ce3> )

Article Best Fonts for UI Design, I use Daily - Best Typefaces and Font Resources for UI Designers ( <https://uxplanet.org/best-fonts-for-ui-design-i-use-daily-4a7bcffb966c> )

YouTube DesignerUp: Best Practices for Choosing Fonts and Font Pairing in UI and Web Design ( <https://www.youtube.com/watch?v=30IoKUGaPWQ> )

YouTube DesignerUp: The Difference Between Fonts, Typefaces and Typography for UI Designers ( <https://www.youtube.com/watch?v=OKGTkLqgm58> )

YouTube DesignerUp: Choosing and Pairing Fonts - UI Design ( <https://www.youtube.com/watch?v=uWCWTq1cPW0> )

YouTube Mizko: The ONLY 8 Fonts UI Designers Need. Forget The Rest. ( <https://www.youtube.com/watch?v=mEAmAFgzQd4> )

## Responsive Design:

Responsive web design (RWD) or responsive design is an approach to web design that aims to make web pages render well on a variety of devices and window or screen sizes from minimum to maximum display size to ensure usability and satisfaction.

Adjusting your pages to the user's screen size

Learning about Media queries

Knowing the concept of Mobile first

### Contents

WebSite W3Schools: HTML Responsive Web Design ( [https://www.w3schools.com/html/html\\_responsive.asp](https://www.w3schools.com/html/html_responsive.asp) )

WebSite MDN Web Docs: Responsive design ( [https://developer.mozilla.org/en-US/docs/Learn/CSS/CSS\\_layout/Responsive\\_Design](https://developer.mozilla.org/en-US/docs/Learn/CSS/CSS_layout/Responsive_Design) )

Article Responsive Web Design: What It Is And How To Use It ( <https://www.smashingmagazine.com/2011/01/guidelines-for-responsive-web-design/> )

Article The Beginner's Guide to Responsive Web Design (Code Samples & Layout Examples) ( <https://kinsta.com/blog/responsive-web-design/> )

YouTube Jesse Showalter: Responsive Web Design | 10 Basics ( <https://www.youtube.com/watch?v=zF6VSKy4Slc> )

YouTube freeCodeCamp.org: Introduction To Responsive Web Design - HTML & CSS Tutorial ( <https://www.youtube.com/watch?v=srvUrASNj0s> )

YouTube Web Dev Simplified: Learn CSS Media Query In 7 Minutes ( <https://www.youtube.com/watch?v=yU7jJ3NbPdA> )



## Android - Accessibility:

Web accessibility is the elimination of barriers on the web. The concept assumes that websites and applications are designed so that all people can effectively perceive, understand, navigate, and interact with pages, including people with accessibility needs, such as those with vision impairments, color blindness, hearing impairments, motor impairments, cognitive disabilities, and many other types of disabilities.

Increasing the visibility of text

Using large, simple controls

Describing each UI element

### Contents

Android Documentation:	
WebSite	Make apps more accessible ( <a href="https://developer.android.com/guide/topics/ui/accessibility/apps">https://developer.android.com/guide/topics/ui/accessibility/apps</a> )
Android Documentation:	
WebSite	Principles for improving app accessibility ( <a href="https://developer.android.com/guide/topics/ui/accessibility/principles">https://developer.android.com/guide/topics/ui/accessibility/principles</a> )
Android Documentation:	
WebSite	Test your app's accessibility ( <a href="https://developer.android.com/guide/topics/ui/accessibility/testing">https://developer.android.com/guide/topics/ui/accessibility/testing</a> )
YouTube	Android Developers: Accessibility on Android ( <a href="https://www.youtube.com/playlist?list=PLWz5rJ2EKKc8OENfLdh3zM5T6IRdIVYKj">https://www.youtube.com/playlist?list=PLWz5rJ2EKKc8OENfLdh3zM5T6IRdIVYKj</a> )
WebSite	Slides about Android Development - by Moro ( <a href="https://github.com/gabrielbmoro/slides-about-android-development/blob/main/README.md#accessibility-%EF%B8%8F">https://github.com/gabrielbmoro/slides-about-android-development/blob/main/README.md#accessibility-%EF%B8%8F</a> )

## Figma - Fundamentals:

Figma is a collaborative web application for interface design. The feature set of Figma focuses on user interface and user experience design, with an emphasis on real-time collaboration, utilising a variety of vector graphics editor and prototyping tools.

Creating page layouts and components

### Contents

Article	Getting Started with Figma ( <a href="https://levelup.gitconnected.com/getting-started-with-figma-637f2c868017">https://levelup.gitconnected.com/getting-started-with-figma-637f2c868017</a> )
Article	Introducing Figma to React ( <a href="https://medium.com/figma-design/introducing-figma-to-react-d2d545cba3cc">https://medium.com/figma-design/introducing-figma-to-react-d2d545cba3cc</a> )
YouTube	Adrian Twarog: Figma Crash Course ( <a href="https://www.youtube.com/watch?v=lg7w3Ntfqy0">https://www.youtube.com/watch?v=lg7w3Ntfqy0</a> )
YouTube	AJ&Smart: Figma UI Design Tutorial - Get Started in Just 24 Minutes ( <a href="https://www.youtube.com/watch?v=FTFaQWZBqQ8">https://www.youtube.com/watch?v=FTFaQWZBqQ8</a> )
YouTube	Jesse Showalter: Intro to Figma - Beginners guide to Figma Basics ( <a href="https://www.youtube.com/watch?v=jk1T0CdLxwU">https://www.youtube.com/watch?v=jk1T0CdLxwU</a> )

YouTube Figma: Figma Tutorial - Components - The Basics ( <https://www.youtube.com/watch?v=k74IrUNaJVk> )

YouTube Figma: Figma Components 101 (<https://www.youtube.com/watch?v=jk1T0CdLxwU>)

CourseLearnUX: Figma Course (<https://learnux.io/course/figma>)

---

TechGuide - Alura  
Alura, PM3 e FIAP  
O Techguide.sh é um projeto open source