



# Curso: Especializado Desenvolvimento de Aplicações para Android

**Duração:** 84h

**Área formativa:** Programação Web & Mobile

---

## Sobre o curso

Desde o lançamento dos smartphones no mercado, é crescente a procura por profissionais que dominem a programação para Android – principalmente, por empresas que se dedicam em exclusivo à produção das aplicações Android. O desenvolvimento de uma aplicação para Web ou Mobile pode, à primeira vista, apresentar algumas semelhanças no entanto, os desafios são distintos. O número de profissionais qualificados para o desenvolvimento de aplicações mobile – Android – continua insuficiente e não acompanha a procura existente no mercado, tornando-se numa excelente oportunidade de carreira.

Este curso tem como objetivos:

Dotar os participantes de conhecimentos e competências necessários para o desenvolvimento de aplicações para Android, recorrendo a programação em Java.

---

## Destinatários

Todos os interessados – maioritariamente programadores – em adquirir e/ou aprofundar conhecimentos de desenvolvimento de aplicações para Android.

---

## Pré-requisitos

Conhecimentos e experiência em qualquer linguagem de programação.

---

## Metodologia

Presencial ou live training.

---

# Programa

- Programação Java para Android (24h)
- Desenvolvimento de Aplicações para Android (30h)
- Workshop Kotlin (9h)
- Projeto (21h)

## Programação Java para Android

- Características da Linguagem
- Java Technology Overview
- Paradigma de Orientação a Objectos
- Messaging
- Desenvolvimento de Aplicações Java
- Introdução à Linguagem Java
- Instanciação
- Variáveis e Tipos de Dados
- Operadores e Expressões
- Arrays
- Controlo de Fluxo
- String e StringBuffer
- Herança
- Framework Java
- Qualificadores e Modificadores de Acesso
- Polimorfismo
- Tratamento de Excepções
- Características da Linguagem
  - Java Virtual Machine
  - Interpretada
  - Portabilidade
  - Multithreaded
  - Carga Dinâmica de Código
  - Orientação a Objectos
  - Java Technology Overview
- Java SE, EE and ME
  - Java Virtual Machine (JVM)
  - Software Based Technology
  - Java Run-Time Environment (JRE)
  - Garbage Collector
- Paradigma de Orientação a Objectos
  - Abstracção
  - Encapsulamento de Atributos e Comportamentos
  - Usar o UML
  - Exercício: Criar dois diagramas de UML
- Messaging
  - Definição de métodos e chamadas
  - Parâmetros/Argumentos
  - Retorno
- Desenvolvimento de Aplicações Java
  - Ferramentas de Desenvolvimento Java
  - Criação de ficheiros JAR

- Escrever, compilar e executar aplicações Java
- Introdução à Linguagem Java
  - Sintaxe e palavras-chave da linguagem
  - Identificadores e convenções da linguagem
- Instanciação
  - Object Blueprint
  - Classes e Objectos Definição
  - Declaração de Classes
  - Instâncias
- Variáveis e Tipos de Dados
  - Atribuição e declaração de variáveis
  - Tipos de dados Primitivos e de Referência/Objetos
  - Casting de tipos de dados
  - Escopo de variáveis
- Operadores e Expressões
  - Funcionalidade
  - Precedência
- Arrays
  - Definição de Arrays
  - Declaração, atribuição e instanciação de Array
  - Atribuição e leitura de índices
  - Exercício: Array simples
- Controlo de Fluxo
  - Repetição
  - Decisão
- String e StringBuffer
  - String
  - StringBuilder
  - Exercício: String &amp;amp; StringBuilder na API
- Herança
  - Hierarquia da Herança
  - SubClass e SuperClass
  - Re-definição de Métodos
  - Dynamic Method Binding
- Framework Java
  - Pacotes de Classes
  - Application Programming Interface (API)
  - Documentação da API
- Qualificadores e Modificadores de Acesso
  - Para: Classes, Variáveis e Metodos
  - Interface de Objecto pública e implementação privada
- Polimorfismo
  - Variáveis, parâmetros e arrays polimorficos
- Tratamento de Excepções
  - Objectos de Excepção
  - try/catch/finally statement
  - Javas Catch ou Specify Requirement
  - Collections
  - Vector, Enumeration, List etc.
  - Input Output
  - Input e Output Streams

- Leitura e Escrita local e remota
- Persistência de Objeto Serialização

## **Desenvolvimento de Aplicações para Android**

- Introdução
  - Ferramentas - Android Studio e Emuladores
  - Framework
  - Versões da plataforma
  - Processo de desenvolvimento
  - Visão geral dos componentes principais
- Activities
  - Classes Activity e Context
  - Ciclo de vida da Activity
  - Android Monitor (logcat e Monitors)
  - Interface gráfica - Layout
  - Associar um Layout a uma Activity
  - Ficheiro Manifesto
  - Deploy da aplicação para o dispositivo
- Listeners
  - Ligação entre a Activity e os componentes do Layout
  - Introdução à biblioteca ButterKnife
  - Introdução às dependências (gradle e maven)
  - Toast
  - OnClickListener, addTextChangedListener, entre outros
  - Recriação da Activity
  - Bundle
- View e Layouts
  - View e ViewGroup
  - Propriedades das Views
  - LinearLayout e RelativeLayout
  - Orientação do ecrã (portrait vs landscape)
  - Tipos de medidas (px, dp, dpi, sp)
  - Views customizadas
  - Drawables e colors
  - Styles
- Fragmentos
  - Classe Fragment
  - Ciclo de vida do Fragmento
  - Fragment Manager
  - Adicionar e remover Fragmentos a uma Activity
  - Fragment Transactions
  - Comunicação entre um Fragmento e Activity
  - Diálogos - DialogFragment
- Intent
  - Iniciar componentes Android
  - Tipos de Intent (explícitos vs implícitos)
  - Passagem de informação entre componentes Android
  - Intent Filters
  - Interface Parcelable
  - Tratamento das Runtime Permissions

- **Adapters**
  - ArrayAdapter
  - ListView
  - Padrão View Holder
  - RecyclerView
- **BroadcastReceivers**
  - Subscrição estática vs Subscrição dinâmica
  - Recepção e envio de broadcasts
  - LocalBroadcastManager
- **Operações em background**
  - AsyncTask
  - UIThread vs Background Threads
  - IntentService
  - AsyncTask vs IntentService
- **Persistência local de Dados**
  - Shared Preferences
  - Internal Storage vs External Storage
  - Classe Application
  - Base de dados SQLite

## **Workshop Kotlin**

- **Características da Linguagem**
  - Tipos Básicos
  - Packages e Imports
  - Control Flow
  - Returns e Jumps
  - Nullables
  - Opcionais
  - Overloading de operadores
- **Classes e Objetos**
  - Classes e hierarquia
  - Propriedades e campos
  - Interfaces
  - Data classes
- **Funções e Lambdas**
  - Funções
  - Lambdas
  - Funções de extensão

## **Projeto**