

## Programação Java para Android

- **Formato do curso:** Presencial e Live Training
- **Localidade:** Porto
- **Data:** 24 Jan. 2019 a 05 Fev. 2019
- **Preço:** 775€
- **Horário:** Pós-laboral e Sábados - 3ª e 5ª, das 18h30 às 21h30 + Sábados das 10h00 às 17h00
- **Nível:** Intermédio
- **Duração:** 24 horas

A linguagem Java é a linguagem que tradicionalmente tem sido utilizada para a criação de apps nativas para o Android. Tem também uma utilização universal, nomeadamente na criação de aplicações empresariais e no desenvolvimento de sites Web.

### Destinatários

Destina-se a programadores interessados em adquirir/aprofundar conhecimentos no desenvolvimento de aplicações para Android recorrendo à tecnologia Java.

### Pré-requisitos

Este curso apresenta como pré-requisito gosto e apetência pelo desenvolvimento de apps para equipamentos móveis. Conhecimentos de algoritmia e experiência em linguagens de programação.

### Objectivos

O curso de programação Java para Android tem como objetivo dotar os formandos dos conhecimentos teóricos e práticos necessários para utilizar a linguagem Java para a criação de aplicações.

### Metodologia

Presencial ou live training.

---

# Programa

## **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**

- Abstração
- 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; 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