

# Compilación y Deployment de PAE

## Compilación local

### Requisitos previos

- **Android Studio:** 2023.2 o superior
- **JDK:** 17
- **SDK:** compileSdk 36, minSdk 27
- **Gradle:** 8.0+ (incluido en el proyecto)

### Configuración inicial

#### 1. Clonar repo y configurar

```
git clone <repo-url>
cd PAE

# Crear local.properties (si no existe)
echo "sdk.dir=/path/to/Android/sdk" > local.properties
```

#### 2. Descargar dependencias

```
./gradlew build --dry-run
# 0
./gradlew downloadDependencies
```

## Compilación por módulo

### Compilar Machine

```
# Debug (desarrollo)
./gradlew :Machine:assembleDebug

# Release (producción)
./gradlew :Machine:assembleRelease
```

### Compilar RutaPAE

```
# Debug
./gradlew :RutaPAE:assembleDebug
```

```
# Release
./gradlew :RutaPAE:assembleRelease
```

## Compilar librerías

```
./gradlew :MachineDomain:build
./gradlew :MachineData:build
./gradlew :Contract:build
# etc.
```

## Compilación completa

```
# Build de todo el proyecto
./gradlew build

# Build con test
./gradlew build connectedCheck

# Clean + build
./gradlew clean build
```

## Configuración de compilación

### Build.gradle (root)

```
plugins {
    id 'com.android.application' version '...'
    kotlin 'org.jetbrains.kotlin.android' version '...'
}

allprojects {
    repositories {
        google()
        mavenCentral()
        jitpack { url "https://jitpack.io" }
    }
}
```

### Build.gradle (Machine)

```
android {
    compileSdk 36
}
```

```
defaultConfig {
    applicationId "co.ada.paemachine"
    minSdk 27
    targetSdk 36
    versionCode 9
    versionName "0.9.0"

    javaCompileOptions {
        annotationProcessorOptions {
            includeCompileClasspath true
        }
    }
}

compileOptions {
    sourceCompatibility = JavaVersion.VERSION_17
    targetCompatibility = JavaVersion.VERSION_17
}

kotlinOptions {
    jvmTarget = '17'
}

buildFeatures {
    compose = true
    buildConfig = true
}

packagingOptions {
    resources {
        excludes += '/META-INF/{AL2.0,LGPL2.1}'
    }
}

dependencies {
    implementation project(':MachineDomain')
    implementation project(':MachineData')
    implementation project(':Contract')
    implementation project(':Core')
    // ... más dependencias
}
```

## Firma de aplicación (Release)

### Configuración de keystore

## 1. Crear keystore (primera vez)

```
cd keystore/  
  
# Machine  
keytool -genkey -v -keystore machine-release.jks \  
-keyalg RSA -keysize 2048 -validity 10000 \  
-alias machine-key \  
-dname "CN=Ada,O=Ada,L=Bogota,ST=DC,C=CO"  
  
# RutaPAE  
keytool -genkey -v -keystore rutapae-release.jks \  
-keyalg RSA -keysize 2048 -validity 10000 \  
-alias rutapae-key \  
-dname "CN=Ada,O=Ada,L=Bogota,ST=DC,C=CO"
```

## 2. Configurar en build.gradle

```
android {  
    signingConfigs {  
        release {  
            storeFile file("../keystore/machine-release.jks")  
            storePassword System.getenv("KEYSTORE_PASS") ?: ""  
            keyAlias "machine-key"  
            keyPassword System.getenv("KEY_PASS") ?: ""  
        }  
    }  
  
    buildTypes {  
        release {  
            signingConfig signingConfigs.release  
            minifyEnabled true  
            proguardFiles getDefaultProguardFile('proguard-android-optimize.txt'), 'proguard-rules.pro'  
        }  
    }  
}
```

## 3. Variables de entorno

```
export KEYSTORE_PASS="tu_keystore_password"  
export KEY_PASS="tu_key_password"  
  
./gradlew :Machine:assembleRelease
```

## Verificar firma

```
# Listar contenido de keystore
keytool -list -v -keystore keystore/machine-release.jks

# Verificar firma de APK
jarsigner -verify -verbose build/outputs/apk/release/Machine-release.apk

# Inspeccionar certificado
zipinfo -1 build/outputs/apk/release/Machine-release.apk | grep META-INF/
```

## Optimización de Release

### ProGuard rules

**ProGuard** optimiza y protege el código:

```
# machine/proguard-rules.pro

# Serialización Kotlin
-keepclassmembers class co.ada.contract.models.** {
    public <init>(...);
    public <methods>;
}

-keepclasshierarchy class co.ada.contract.models.**

# Enums
-keepclassmembers enum * {
    public static **[] values();
    public static ** valueOf(java.lang.String);
}

# Room (si se usara)
-keep class **.BuildConfig { *; }

# Composable
-keep @androidx.compose.runtime.Composable class ** { *; }
```

### R8 (reemplazo moderno de ProGuard)

```
android.enableR8 = true
android.enableR8.fullMode = true # modo agresivo
```

# Build script automatizado

## build-release.sh (Linux/Mac)

```
#!/bin/bash

set -e

echo "=== Build PAE Release ==="

# Limpiar
./gradlew clean

# Compilar Machine
echo "Building Machine..."
./gradlew :Machine:assembleRelease

# Compilar RutaPAE
echo "Building RutaPAE..."
./gradlew :RutaPAE:assembleRelease

# Generar APKs con alineación
./gradlew :Machine:alignRelease
./gradlew :RutaPAE:alignRelease

echo "=== Build completado ==="
echo "Machine: build/outputs/apk/release/Machine-release.apk"
echo "RutaPAE: build/outputs/apk/release/RutaPAE-release.apk"
```

## build-release.ps1 (Windows)

```
Write-Host "=== Build PAE Release ==="

# Limpiar
& .\gradlew.bat clean

# Machine
Write-Host "Building Machine..."
& .\gradlew.bat :Machine:assembleRelease

# RutaPAE
Write-Host "Building RutaPAE..."
& .\gradlew.bat :RutaPAE:assembleRelease

Write-Host "=== Build completado ==="
```

# Deployment

## Instalación en dispositivo

### Debug APK

```
adb install build/outputs/apk/debug/Machine-debug.apk
adb install build/outputs/apk/debug/RutaPAE-debug.apk
```

### Release APK

```
adb install build/outputs/apk/release/Machine-release.apk
adb install build/outputs/apk/release/RutaPAE-release.apk
```

### Reinstalar

```
adb install -r build/outputs/apk/release/Machine-release.apk
```

### Desinstalar

```
adb uninstall co.ada.paemachine
adb uninstall co.ada.rutapae
```

## Ver logs de instalación

```
adb logcat -s PackageManager
```

## CI/CD (Integración continua)

### GitHub Actions (ejemplo)

```
# .github/workflows/build.yml
name: Build PAE

on: [push, pull_request]

jobs:
  build:
    runs-on: ubuntu-latest
    steps:
```

```
- uses: actions/checkout@v3
- name: Set up JDK 17
  uses: actions/setup-java@v3
  with:
    java-version: '17'
    distribution: 'temurin'
- name: Build Machine
  run: ./gradlew :Machine:assembleRelease
- name: Build RutaPAE
  run: ./gradlew :RutaPAE:assembleRelease
- name: Upload artifacts
  uses: actions/upload-artifact@v3
  with:
    name: apks
    path: build/outputs/apk/release/
```

## Testing en CI/CD

```
# Unit tests
./gradlew test

# Android tests (requiere emulador)
./gradlew connectedAndroidTest

# Lint
./gradlew lint

# Lint + tests + build
./gradlew build connectedCheck
```

## Versionado

### Versión de Machine

- **Pattern:** MAJOR.MINOR.PATCH (e.g., 0.9.0)
- **Current:** 0.9.0 (versión pre-release)
- **versionCode:** incremental (9)

### Versión de RutaPAE

- **Pattern:** MAJOR.MINOR.PATCH (e.g., 0.4.0)
- **Current:** 0.4.0 (versión pre-release)
- **versionCode:** incremental (4)

## Actualizar versión

```
// Machine/build.gradle.kts
android {
    defaultConfig {
        versionCode 10
        versionName "0.10.0"
    }
}
```

## Troubleshooting de build

**Problema: "org.gradle.api.GradleException: org.gradle.workers.internal.DefaultWorkerExecutor\$WorkerExecutionException"**

**Solución:**

```
./gradlew clean --build-cache
./gradlew build
```

**Problema: "Duplicate class kotlin.concurrent.LockedWrapper"**

**Solución:** eliminar dependencias duplicadas de Kotlin

```
configurations {
    all*.exclude module: 'kotlin-stdlib'
}
```

**Problema: "Could not resolve <dependency>"**

**Solución:** actualizar repositorios

```
./gradlew :dependencies
./gradlew clean build --refresh-dependencies
```

**Problema: "Task :app:linDebugJavaWithJavac FAILED"**

**Solución:** versión de JDK incorrecta

```
java -version
# debe ser 17+
```

# Cache y performance

## Limpiar cache

```
# Gradle cache
rm -rf ~/.gradle/caches

# Build cache
./gradlew cleanBuildCache

# Android Studio cache
rm -rf ~/.android/cache
```

## Compilación paralela

```
# build.gradle
org.gradle.parallel=true
org.gradle.workers.max=8
```

## Daemon gradle

```
# Habilitar daemon
org.gradle.daemon=true

# Desabilitar si hay problemas
./gradlew --no-daemon build
```

## Monitoreo de build

### Análisis de tiempos

```
./gradlew build --profile

# Genera HTML en build/reports/profile/
```

### Lint report

```
./gradlew lint

# Report en build/reports/lint-results.html
```

## Dependency report

```
./gradlew dependencies
```

```
# Genera árbol de dependencias
```

From:

<http://wiki.adacsc.co/> - **Wiki**

Permanent link:

<http://wiki.adacsc.co/doku.php?id=ada:howto:sicoferp:factory:new-migracion-sicoferp:build-and-deployment>

Last update: **2026/04/07 19:57**

