

App Rastrea (Flutter)

Todos los repositorios se guardan en:

C:/overskull/

? 1. Versiones necesarias para este proyecto

Herramienta	Versión
Flutter	3.19.6
Dart	3.3.x (incluido en Flutter 3.19.6)
JDK	Amazon Corretto 17
Gradle	Automático según Flutter

Este proyecto **NO usa Java 11**. Usar exclusivamente **Java 17**.

? 2. Descargar Flutter (versión obligatoria del proyecto)

Descargar Flutter **3.19.6 (stable)** desde:

<https://docs.flutter.dev/release/archive>

Guardar todas las versiones de Flutter en:

C:/overskull/SDK/

Ejemplo de ruta correcta:

C:/overskull/SDK/flutter_windows_3.19.6-stable

No usar Flutter global del sistema. El proyecto apunta explícitamente a su versión mediante `.vscode/settings.json`.

? 3. Instalar JDK 17 (Amazon Corretto)

Descargar desde:

<https://docs.aws.amazon.com/corretto/latest/corretto-17-ug/downloads-list.html>

☐ Guardar todas las versiones de Java en:

```
C:/overskull/java/
```

Ejemplo:

```
C:/overskull/java/amazon-corretto-17
```

? 4. Clonar el proyecto

```
git clone git@storegit.overskull.com:shalom-over-app/shalom_rastrea.git
cd shalom_rastrea
```

? 5. build.gradle (revisión inicial necesaria para arrancar el proyecto)

Ruta:

```
C:/overskull/shalom_rastrea/android/build.gradle
```

Ejemplo de referencia:

```
plugins {
    id "com.android.application"
    id "kotlin-android"
    id "dev.flutter.flutter-gradle-plugin"
}

def localProperties = new Properties()
def localPropertiesFile = rootProject.file('local.properties')
if (localPropertiesFile.exists()) {
    localPropertiesFile.withReader('UTF-8') { reader ->
        localProperties.load(reader)
    }
}

def flutterVersionCode = localProperties.getProperty('flutter.versionCode')
if (flutterVersionCode == null) {
```

```
flutterVersionCode = '1'
}

def flutterVersionName = localProperties.getProperty('flutter.versionName')
if (flutterVersionName == null) {
    flutterVersionName = '1.0'
}

def keystoreProperties = new Properties()
def keystorePropertiesFile = rootProject.file('key.properties')
if (keystorePropertiesFile.exists()) {
    keystoreProperties.load(new FileInputStream(keystorePropertiesFile))
}

android {
    compileSdkVersion 34

    namespace "pe.com.shalom.overskull"
    compileSdkVersion flutter.compileSdkVersion
    //ndkVersion flutter.ndkVersion
    ndkVersion "25.2.9519653"

    compileOptions {
        sourceCompatibility JavaVersion.VERSION_1_8
        targetCompatibility JavaVersion.VERSION_1_8
    }

    kotlinOptions {
        jvmTarget = '1.8'
    }

    sourceSets {
        main.java.srcDirs += 'src/main/kotlin'
    }

    // ndkVersion "25.1.8937393"

    defaultConfig {
        // TODO: Specify your own unique Application ID
        (<https://developer.android.com/studio/build/application-id.html>).
```

```
    applicationId "pe.com.shalom.overskull"
    // You can update the following values to match your application needs.
    // For more information, see: <https://docs.flutter.dev/deployment/android#reviewing-
the-gradle-build-configuration>.
    minSdkVersion 24
    targetSdkVersion 35
    versionCode 284
    versionName "2.8.4"
    multiDexEnabled true
}
signingConfigs {
    release {
        keyAlias keystoreProperties['keyAlias']
        keyPassword keystoreProperties['keyPassword']
        storeFile keystoreProperties['storeFile'] ? file(keystoreProperties['storeFile']) :
null
        storePassword keystoreProperties['storePassword']
    }
}

buildTypes {
    release {
        // TODO: Add your own signing config for the release build.
        // Signing with the debug keys for now, so `flutter run --release` works.
        minifyEnabled false // Asegúrate de que está en false
        shrinkResources false
        signingConfig signingConfigs.release
    }
}

packagingOptions {
    exclude 'META-INF/DEPENDENCIES'
    jniLibs {
        useLegacyPackaging = false
    }
    doNotStrip "**/*.so"
}
}
```

```
flutter {
    source '../..'
}

dependencies {
    implementation 'com.google.android.material:material:1.4.0'
    implementation "org.jetbrains.kotlin:kotlin-stdlib-jdk7:$kotlin_version"
    testImplementation 'junit:junit:4.12'
    androidTestImplementation 'com.android.support.test:runner:1.0.2'
    androidTestImplementation 'com.android.support.test.espresso:espresso-core:3.0.2'
    implementation 'com.android.support:support-fragment:28.0.0'
    implementation 'com.google.http-client:google-http-client:1.28.0'
    implementation 'com.google.http-client:google-http-client-android:1.28.0'
    implementation 'com.google.http-client:google-http-client-jackson2:1.28.0'
    implementation 'com.fasterxml.jackson.core:jackson-core:2.9.8'
    implementation 'org.slf4j:slf4j-log4j12:1.7.25'
    implementation 'com.google.guava:guava:27.0.1-android'
    implementation fileTree(dir: "libs", include: ["*.jar", "*.aar"])
}
```

☐ Verificar principalmente:

- Uso de **Java 17**
- Compatibilidad de `compileSdkVersion` y `targetSdkVersion`
- Plugins de Android y Flutter

☐ Si este archivo tiene errores, la app no podrá compilar.

?? 6. Configurar JDK en gradle.properties

Editar:

```
C:/overskull/shalom_rastrea/android/gradle.properties
```

Contenido:

```
org.gradle.jvmargs=-XX:MaxHeapSize=4048m
android.useAndroidX=true
android.enableJetifier=true
org.gradle.java.home=C:/Program Files/Amazon Corretto/jdk17.0.17_10
```

☐ Cambiar la ruta según donde se guardó el JDK.

? 7. Configurar la versión de Flutter usada por el proyecto

Editar o crear:

```
C:/overskull/shalom_rastrea/.vscode/settings.json
```

```
{
  "dart.flutterSdkPath": "C:/overskull/SDK/flutter_windows_3.19.6-stable",
  "java.configuration.updateBuildConfiguration": "interactive"
}
```

?? 8. Configurar JDK en launch.json (VSCode)

Editar o crear:

```
C:/overskull/shalom_rastrea/.vscode/launch.json
```

```
{
  "version": "0.2.0",
  "configurations": [
    {
      "name": "Flutter Run Without Debug",
      "request": "launch",
      "type": "dart",
      "flutterMode": "debug",
      "noDebug": true,
      "program": "lib/main.dart",
      "env": {
        "JAVA_HOME": "C:/overskull/java/amazon-corretto-17"
      }
    }
  ]
}
```

? 9. Configuración adicional para Linux / multisistema

Editar:

```
android/gradle.properties
```

Agregar:

```
org.gradle.java.home=/home/usuario/java/amazon-corretto-17
```

? 10. Limpiar el proyecto

```
cd android  
gradlew clean
```

? 11. Descargar dependencias

```
flutter pub get
```

? 12. Ejecutar el proyecto

```
flutter run
```

? Notas importantes

- Este proyecto usa **Java 17**. No reutilizar configuraciones de Java 11.
- Siempre validar `gradle.properties` y `.vscode/settings.json` antes de compilar.
- Guía oficial Overskull - Flutter 3.19.6 + JDK 17.

Revisión #5

Creado 2025-12-13 10:52:25 -05 por Daisy

Actualizado 2025-12-13 11:33:02 -05 por Daisy