

# App Familia (Flutter)

Todos los repositorios se guardan en:

C:\\overskull\\

## ? 1. Versiones necesarias para este proyecto

Herramienta	Versión
Flutter	3.29.3
Dart	3.3.x (incluido en Flutter 3.19.6)
JDK	Amazon Corretto 11
Gradle	Automático según Flutter

Leer este punto antes de continuar.

Este proyecto **NO funciona** con Java 17 ni con Flutter global.

## ? 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:

C:\\overskull\\SDK\\flutter\_windows\_3.29.3-stable

No usar Flutter global del sistema.

## ? 3. Instalar JDK 11 (Amazon Corretto)

Descargar desde:

<https://docs.aws.amazon.com/corretto/latest/corretto-11-ug/downloads-list.html>

Guardar todas las versiones de Java en:

```
C:\\overskull\\java\\
```

Ejemplo:

```
C:\\overskull\\java\\amazon-corretto-11
```

## ? 4. Clonar el proyecto

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

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

Ruta:

```
C:\\overskull\\shalom_familia\\android\\app\\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 31

    namespace "com.overskull.shalom.familia"
    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'
    }

    defaultConfig {
        // TODO: Specify your own unique Application ID
        (<https://developer.android.com/studio/build/application-id.html>).
        applicationId "com.overskull.shalom.familia"
        // 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 148
    versionName "1.4.8"
  }
  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.
      signingConfig signingConfigs.release
    }
  }

  packagingOptions {
    jniLibs {
      useLegacyPackaging = false
    }
    doNotStrip "**/*.so"
  }
}

flutter {
  source '../..'
}

dependencies {}
```

☐ Si este archivo tiene errores, la app **no compila**.

---

## ?? 6. Configurar JDK en gradle.properties

Editar:

```
C:\\overskull\\shalom_familia\\android\\gradle.properties
```

```
org.gradle.jvmargs=-Xmx4608m
android.useAndroidX=true
android.enableJetifier=true
org.gradle.java.home=C:/Program Files/Amazon Corretto/jdk11.0.29_7
android.enableAppCompileTimeRClass=true
android.bundle.enableUncompressedNativeLibs=false
```

---

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

```
C:\\overskull\\shalom_familia\\.vscode\\settings.json
```

```
{
  "dart.flutterSdkPath": "C:/SDK/flutter_windows_3.19.6-stable"
}
```

---

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

```
C:\\overskull\\shalom_familia\\.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-11"
      }
    }
  ]
}
```

```
    }  
  }  
]  
}
```

## ? 9. Configuración adicional para Linux / multisistema

```
android/gradle.properties
```

```
org.gradle.java.home=/home/usuario/java/amazon-corretto-11
```

## ? 10. Limpiar el proyecto

```
cd android  
./gradlew clean
```

## ? 11. Descargar dependencias

```
flutter pub get
```

## ?? 12. Modificaciones obligatorias

### pubspec.yaml

```
pdfx: ^2.6.0  
flutter_slidable: ^3.1.2
```

### Archivo Términos y Condiciones

Ruta:

```
C:\\overskull\\shalom_familia\\lib\\src\\Pages\\TerminosYCondiciones\\termino_View.dart
```

Cambiar:

□

```
primary: Color.fromRGBO(107, 107, 107, 1),
```

✓ por:

```
foregroundColor: Color.fromRGBO(107, 107, 107, 1),
```

☐ Este cambio es obligatorio para compatibilidad con Flutter 3.19.6.

---

## ? 13. Listo! Ahora puedes ejecutar el proyecto

```
flutter run
```

### ? Notas importantes

- Este proyecto usa **Java 11 (Amazon Corretto)**.
  - ☐ No usar Java 17 ni versiones superiores.
- Verificar siempre que estas rutas coincidan:
  - `.vscode/settings.json` → versión correcta de Flutter
  - `android/gradle.properties` → ruta correcta del JDK
- El archivo `build.gradle` **no debe ser modificado** sin revisar compatibilidad con:
  - `compileSdkVersion`
  - `ndkVersion 25.2.9519653`
- La modificación en **Términos y Condiciones** es **obligatoria** para evitar errores de compilación en Flutter 3.19.6.
- Si el proyecto no compila a la primera:

```
flutter clean  
flutter pub get
```

---

Revisión #4

Creado 2025-12-12 17:19:11 -05 por Daisy

Actualizado 2026-01-07 12:10:28 -05 por Edson Alvarado