

Crear y publicar librerías

Descargar estos apuntes [pdf](#) o [html](#)

Migrar proyecto a Android Studio 2024 con Kotlin 2.0.20

1. File → Project Structure → Gradle Versión -> **8.7**
2. Tools → AGP upgrade assistant → **8.6.0**
3. Crear `libs.versions.toml` a la carpeta `gradle`



Aviso

No te olvides de quitar aquellas dependencias que no se usen, **room**, **retrofit**, etc.

```

[versions]
agp = "8.6.0"
kotlin = "2.0.20"
coreKtx = "1.13.1"
junit = "4.13.2"
junitVersion = "1.2.1"
espressoCore = "3.6.1"
lifecycleRuntimeKtx = "2.8.6"
activityCompose = "1.9.2"
composeBom = "2024.09.02"

pmdmIesBalmisVersion = "24.1"
ksp = "2.0.20-1.0.25"
hilt = "2.52"
hiltNavigationCompose = "1.2.0"
composeNavigation = "2.8.1"
room = "2.6.1"

[libraries]
androidx-core-ktx = { group = "androidx.core", name = "core-ktx", version.ref = "coreKtx" }
junit = { group = "junit", name = "junit", version.ref = "junit" }
androidx-junit = { group = "androidx.test.ext", name = "junit", version.ref = "junitVersion" }
androidx-espresso-core = {
    group = "androidx.test.espresso",
    name = "espresso-core",
    version.ref = "espressoCore" }
androidx-lifecycle-runtime-ktx = {
    group = "androidx.lifecycle",
    name = "lifecycle-runtime-ktx",
    version.ref = "lifecycleRuntimeKtx" }
androidx-activity-compose = {
    group = "androidx.activity",
    name = "activity-compose",
    version.ref = "activityCompose" }
androidx-compose-bom = {
    group = "androidx.compose",
    name = "compose-bom",
    version.ref = "composeBom" }
androidx-ui = { group = "androidx.compose.ui", name = "ui" }
androidx-ui-graphics = { group = "androidx.compose.ui", name = "ui-graphics" }
androidx-ui-tooling = { group = "androidx.compose.ui", name = "ui-tooling" }
androidx-ui-tooling-preview = { group = "androidx.compose.ui", name = "ui-tooling-preview" }
androidx-ui-test-manifest = { group = "androidx.compose.ui", name = "ui-test-manifest" }
androidx-ui-test-junit4 = { group = "androidx.compose.ui", name = "ui-test-junit4" }
androidx-material3 = { group = "androidx.compose.material3", name = "material3" }

compose-navigation = {
    group = "androidx.navigation",
    name = "navigation-compose",
    version.ref = "composeNavigation" }

```

```

androidx-hilt-navigation-compose = {
    group = "androidx.hilt",
    name = "hilt-navigation-compose",
    version.ref = "hiltNavigationCompose" }

dagger-hilt-android = {
    group = "com.google.dagger",
    name = "hilt-android",
    version.ref = "hilt" }
dagger-hilt-android-compiler = {
    group = "com.google.dagger",
    name = "hilt-android-compiler",
    version.ref = "hilt" }

androidx-room-ktx = { group = "androidx.room", name = "room-ktx", version.ref = "room" }
androidx-room-compiler = { group = "androidx.room", name = "room-compiler", version.ref = "room" }

pmdm-ies-balmis-components = {
    group = "com.github.pmdmiesbalmis",
    name = "components",
    version.ref = "pmdmIesBalmisVersion" }

[plugins]
android-application = { id = "com.android.application", version.ref = "agp" }
kotlin-android = { id = "org.jetbrains.kotlin.android", version.ref = "kotlin" }
compose-compiler = { id = "org.jetbrains.kotlin.plugin.compose", version.ref = "kotlin" }
devtools-ksp = { id = "com.google.devtools.ksp", version.ref = "ksp" }
com-google-dagger = { id = "com.google.dagger.hilt.android", version.ref = "hilt" }

```

4. Cambiar el contenido de `settings.gradle.kts` por:

```

pluginManagement {
    repositories {
        google {
            content {
                includeGroupByRegex("com\\.android.*")
                includeGroupByRegex("com\\.google.*")
                includeGroupByRegex("androidx.*")
            }
        }
        mavenCentral()
        gradlePluginPortal()
    }
}

dependencyResolutionManagement {
    repositoriesMode.set(RepositoriesMode.FAIL_ON_PROJECT_REPOS)
    repositories {
        google()
        mavenCentral()
        maven {
            url = uri("https://jitpack.io")
        }
    }
}

rootProject.name = "NOMBRE DE MI PROYECTO"
include(":app")

```

5. En `build.gradle.kts`, general a todos los módulos, agregar los plugins:

```

plugins {
    alias(libs.plugins.android.application) apply false
    alias(libs.plugins.kotlin.android) apply false
    alias(libs.plugins.compose.compiler) apply false
    alias(libs.plugins.devtools.ksp) apply false
    alias(libs.plugins.com.google.dagger) apply false
}

```

6. En `build.gradle.kts`, de la aplicación:

- Agregar los plugins:

```

plugins {
    alias(libs.plugins.android.application)
    alias(libs.plugins.kotlin.android)
    alias(libs.plugins.compose.compiler)
    alias(libs.plugins.devtools.ksp)
    alias(libs.plugins.com.google.dagger)
    id("kotlin-parcelize")
}

```

- Eliminar bloque `composeOptions` dentro del bloque `android` ya que el compilador de compose viene ahora incluido en un plugin:

```
composeOptions {  
    kotlinCompilerExtensionVersion = "1.5.0"  
}
```

- En el bloque `dependencies` cambiar todas por:

 **Aviso**

No te olvides de quitar aquellas dependencias que no se usen, **room**, **retrofit**, etc.

```

dependencies {
    implementation(libs.androidx.core.ktx)
    implementation(libs.androidx.lifecycle.runtime.ktx)
    implementation(libs.androidx.activity.compose)
    implementation(platform(libs.androidx.compose.bom))
    implementation(libs.androidx.ui)
    implementation(libs.androidx.ui.graphics)
    implementation(libs.androidx.ui.tooling.preview)
    implementation(libs.androidx.material3)

    implementation(libs.pmdm.ies.balmis.components)

    implementation(libs.dagger.hilt.android)
    implementation(libs.androidx.hilt.navigation.compose)
    ksp(libs.dagger.hilt.android.compiler)
    kspAndroidTest(libs.dagger.hilt.android.compiler)

    implementation(libs.compose.navigation)

    implementation(libs.androidx.room.ktx)
    ksp(libs.androidx.room.compiler)

    implementation(libs.com.squareup.retrofit2.converter.gson)
    implementation(libs.com.squareup.retrofit2.retrofit)
    implementation(platform(libs.com.squareup.okhttp3.okhttp.bom))
    implementation(libs.com.squareup.okhttp3.okhttp)
    implementation(libs.com.squareup.okhttp3.logging.interceptor)

    testImplementation(libs.junit)
    androidTestImplementation(libs.androidx.junit)
    androidTestImplementation(libs.androidx.espresso.core)
    androidTestImplementation(platform(libs.androidx.compose.bom))
    androidTestImplementation(libs.androidx.ui.test.junit4)
    debugImplementation(libs.androidx.ui.tooling)
    debugImplementation(libs.androidx.ui.test.manifest)
}

```

7. Hacer **gradle sync** y hacer el **Scan for deprecations** que propone al migrar la versión de Kotlin.

8. En el archivo `ui.theme.Theme.kt` eliminar el bloque deprecado y **optimizar imports**.

```

...
val view = LocalView.current
if (!view.isInEditMode) {
    SideEffect {
        (view.context as Activity).window.statusBarColor = colorScheme.primary.toArgb()
        ViewCompat.getWindowInsetsController(view)?.isAppearanceLightStatusBars = darkTheme
    }
}
...

```

9. Eliminar paquetes `ui.composables`, `utilities.validacion` y `manejo_errores`. Posteriormente arreglar dependencias con la librería nueva.

- **Replace in Files** (`Ctrl+Shift+H`) todos los imports de imports en el proyecto de `com.pmdm.XXXX.utilities.validacion` y sustituirlos por `com.github.pmdmiesbalmis.composables`.
- Si hay errores de iconos ir a <https://fonts.google.com/icons>:
 - Marcar el switch fill.
 - Buscar el ícono deseado y seleccionar la carpeta Android.
 - Descargar el recurso xml desde la web e incluirlo en la carpeta `res/drawable` y sustituirlo por el correspondiente `import androidx.compose.material.icons.filled.XXXX`.
- Si es un ícono muy común puedes buscarlo en `com.github.pmdmiesbalmis.components.ui.icons.Filled`. Para ello, añade la clase `Filled` y te propondrá importar el paquete `com.github.pmdmiesbalmis.components.ui.icons.Filled`

10. Cuando compile todo. Botón derecho sobre App → Refactor → **Migrate Build Config to Gradle Build Files**. Pulsar **Do Refactor**.