

# Características de AndroidX

¡Saludos! Excelente momento para adquirir nuevos conocimientos con respecto al desarrollo en Android. A continuación te presentamos un resumen de **AndroidX**.

Los **artefactos (artifacts) dentro del espacio de nombres de AndroidX comprenden las librerías de Android Jetpack**. Al igual que la librería de soporte (Support Library), las librerías en el espacio de nombres de Android se envían por separado de la plataforma Android y proporcionan compatibilidad con versiones anteriores de Android.

**AndroidX es una mejora importante de la librería de soporte de Android original**, que ya no se mantiene (es decir, no se le van a hacer nuevas mejoras o actualizaciones). Los paquetes de AndroidX reemplazan completamente la librería de soporte al proporcionar paridad de funciones y nuevas librerías.

Además, **AndroidX incluye las siguientes características:**

- **Todos los paquetes en AndroidX viven en un espacio de nombres consistente** que comienza con la cadena androidX. Los paquetes de la librería de soporte se han mapeado en los correspondientes paquetes androidX. \*. Para obtener una asignación completa de todas las clases antiguas y crear artefactos con las nuevas, consulte la página Refactorización de paquetes.
- A diferencia de la librería de soporte, **los paquetes de AndroidX se mantienen y actualizan por separado**. Los paquetes de AndroidX usan versiones semánticas estrictas, comenzando con la versión 1.0.0. Puede actualizar las librerías de AndroidX en su proyecto de forma independiente.
- **La versión 28.0.0 es la última versión de la librería de soporte**. No habrá más versiones de la librería android.support. Todo el desarrollo de nuevas características estará en el espacio de nombres de AndroidX.

## Usando AndroidX en nuestro proyecto

Si deseamos utilizar las librerías con espacios de nombres de AndroidX en un nuevo proyecto, debemos configurar el SDK de compilación en Android 9.0 (API nivel 28) o superior y establecer los siguientes indicadores de complemento (plugins) de Android Gradle en true dentro de nuestro archivo **gradle.properties**.

- **android.useAndroidX:** cuando este indicador se establece en verdadero, el complemento de Android utiliza la librería de AndroidX adecuada en lugar de la librería de soporte. El indicador es falso de forma predeterminada, si no se especifica.
- **android.enableJetifier:** cuando este indicador se establece en verdadero, el complemento de Android migra automáticamente las librerías de terceros existentes para usar las dependencias de AndroidX, reescribiendo sus archivos binarios. El indicador es falso de forma predeterminada, si no se especifica.

# Comenzando con Android Jetpack

**Jetpack abarca una colección de librerías de Android** que incorporan las mejores prácticas y proporcionan compatibilidad con versiones anteriores en tus aplicaciones de Android.

La guía de Jetpack para la arquitectura de la aplicación proporciona una descripción general de las mejores prácticas y la arquitectura recomendada a tener en cuenta al crear tu aplicación de Android.

Las siguientes secciones cubren cómo se puede comenzar a usar los componentes de Jetpack.

## Usar una librería Jetpack en nuestra aplicación

Todos los componentes de Jetpack están disponibles en el repositorio de Google Maven.

Abre el archivo **build.gradle** para nuestro proyecto y agrega el **repositorio google ()** como se muestra a continuación:

```
allprojects {
    repositories {
        google()
        jcenter()
    }
}
```

Luego podemos agregar los componentes de Jetpack, así como componentes de arquitectura como **LiveData** y **ViewModel** que forman parte de la librería Lifecycles, como se muestra aquí:

```
dependencies {
    def lifecycle_version = "2.0.0"
    implementation
    "androidx.lifecycle:lifecycle-extensions:$lifecycle_version"
    // Optional : Kotlin extension (https://d.android.com/kotlin/ktx)
    implementation "androidx.lifecycle:lifecycle-viewmodel-
    ktx:$lifecycle_version"
    ...
}
```

Muchas librerías Jetpack también proporcionan extensiones **Android KTX** como se muestra arriba en **lifecycle-viewmodel-ktx**. Las **extensiones KTX** se apoyan en la API basada en Java, aprovechando las características de lenguaje específicas de Kotlin.

Para conocer las nuevas versiones de la librería Jetpack, consultemos la página de versiones. Las páginas de referencia de API basadas en Kotlin y Java están disponibles para todas las librerías Jetpack.

# Aprovechemos Jetpack

Las **librerías Jetpack se pueden usar solas o en combinación** para abordar diferentes necesidades en tus aplicaciones. Por ejemplo:

- **WorkManager** para tus necesidades de programación en segundo plano.
- **Room** para la persistencia del almacenamiento de datos.
- **Navigation** para administrar el flujo de navegación de nuestra aplicación.
- **CameraX** para las necesidades de nuestra aplicación de cámara.

Esperamos que hayas disfrutado de algunos aspectos relevantes de AndroidX. Continúa tu proceso de aprendizaje ampliando tus conocimientos.