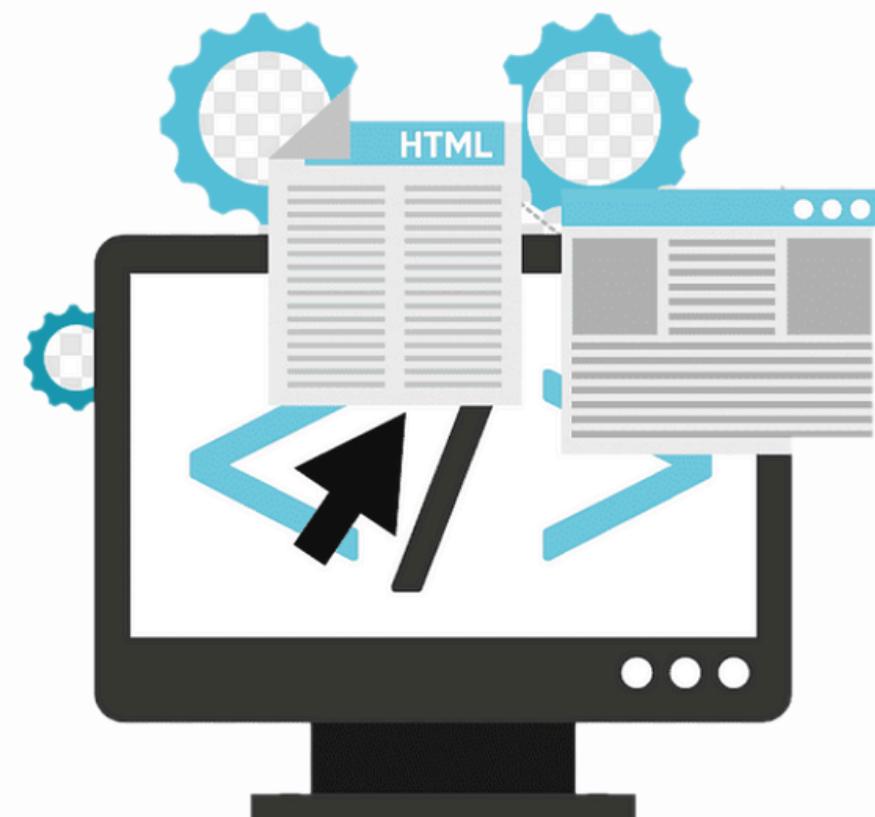


# Herramientas de Empaquetado y Despliegue



Presentado por:

- LABRA MARTÍNEZ ALAN
- MARTÍNEZ MARTÍN OSBALDO
- MUÑOZ ÁLVAREZ GABRIELA SCARLETT
- PEDRAZA CRUZ ODALYS
- ROMERO BARRÓN LUIS FERNANDO

# Índice

- Definición de Empaquetado
- Características del Empaquetado
- Uso de las Herramientas de Empaquetado
- Herramientas de Empaquetado en Flutter
- Ventajas del Empaquetado
- Desventajas del Empaquetado
- Definición de Despliegue
- Uso del Despliegue en Flutter
- Características del Despliegue
- Ventajas del Despliegue
- Desventajas del Despliegue
- Comandos base de Flutter CLI
- Herramientas de Automatización y CI/CD
- Herramientas de Gestión de Entornos y Configuraciones
- Herramientas de Despliegue
- Referencias

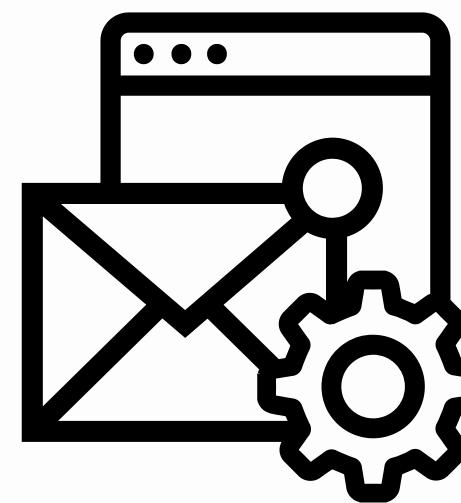


# Introducción

Las herramientas de **empaque y despliegue** son fundamentales en el ciclo de desarrollo de software. Estas herramientas optimizan la distribución y ejecución de aplicaciones, garantizando un proceso eficiente y seguro. En esta presentación, exploraremos definiciones clave, características de herramientas como Flutter CLI y CI/CD, y la importancia de la gestión de entornos. Comprender estas herramientas es vital para estudiantes y desarrolladores en el actual panorama tecnológico.

# ¿Qué es el EMPAQUETADO?

Una herramienta de empaquetado es un software que permite reunir todos los componentes necesarios de un programa (como archivos ejecutables, bibliotecas, dependencias, imágenes y configuraciones) en un solo paquete instalable o ejecutable, listo para su distribución o instalación.



# Características del Empaquetado



## Agrupación de archivos

Reúnen todos los elementos necesarios del programa en un solo paquete.

## Compresión

Reducen el tamaño del paquete para facilitar almacenamiento o envío.

## Gestión de dependencias

Incluyen las librerías necesarias para el funcionamiento del software.

## Portabilidad

Permiten ejecutar o instalar fácilmente la aplicación en distintos sistemas operativos o entornos.

## Uso de las Herramientas de Empaquetado

Permiten incorporar paquetes reutilizables y crear los paquetes finales de la aplicación.

Principal herramienta: Pub, gestor de dependencias que agrega y administra paquetes de código.



# Ventajas de Herramientas de Empaqueado en Flutter

- 01 Compila a código nativo (AOT), ofreciendo alto rendimiento.
- 02 Usa compilación JIT y hot reload, agilizando el desarrollo.
- 03 Un solo código base para Android, iOS, web y escritorio.
- 04 Proceso de empaquetado integrado y simple mediante comandos.
- 05 SDK con widgets personalizables y eficientes.

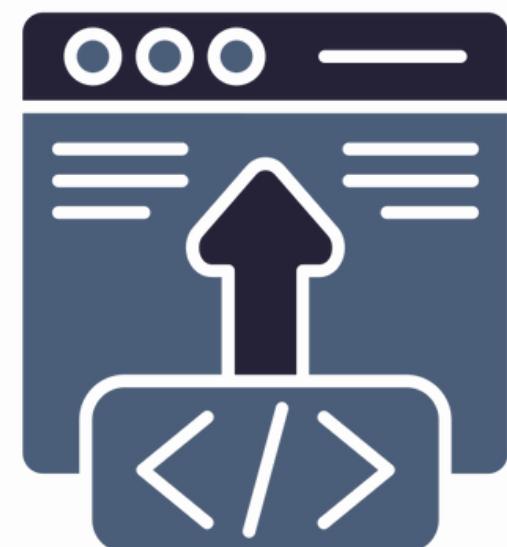
# Desventajas del Empaquetado en Flutter

- 
- 01 Tamaño de archivo mayor debido al motor incluido.
  - 02 Algunas funciones nativas requieren código o plugins específicos.
  - 03 Ecosistema joven con menos librerías que otros frameworks.

Fuente: Vilanou, R. (2025, 30 julio).  
Flutter: ventajas y desventajas del desarrollo mobile. Doonamis.

## ¿Qué es el DESPLIEGUE?

Una herramienta de despliegue automatiza el proceso de trasladar una aplicación desde su entorno de desarrollo hasta su entorno de producción, asegurando que funcione correctamente y con la configuración adecuada.



# Uso del Despliegue en Flutter

El despliegue en Flutter permite crear aplicaciones multiplataforma desde una sola base de código.

Incluye la generación de un “build” (paquete de producción) que se sube a la plataforma correspondiente (Play Store, App Store o servidor web).



# Características del Despliegue

## Automatización

Permiten desplegar aplicaciones sin hacerlo manualmente.

## Gestión de versiones

Controlan qué versión se instala y permiten revertir si hay fallos.

## Configuración de entornos

Ajustan los parámetros del sistema o servidor automáticamente.

## Escalabilidad

Permiten desplegar en varios servidores o máquinas simultáneamente.

# Ventajas de Herramientas de Despliegue

- 01 Interfaz intuitiva que simplifica configuración.
- 02 Flujo de trabajo optimizado con Codemagic.
- 03 Soporte de firma de código e integración con Apple.
- 04 Automatización de tareas repetitivas.
- 05 Reducción de errores y control detallado del proceso.
- 06 Integración con CI/CD y flexibilidad de publicación.

# Desventajas de Herramientas de Despliegue

- 
- 01 Bloqueo de proveedor: Dificulta migrar a otras plataformas.
  - 02 Requiere configuración inicial compleja.
  - 03 Curva de aprendizaje para entender la automatización móvil.

# Comandos Base de Flutter CLI

Son los comandos esenciales para compilar y empaquetar tu aplicación para distintas plataformas.

## **flutter build apk**

Crea un paquete APK  
listo para Google Play  
o instalación directa.

## **flutter build appbundle**

Genera un App  
Bundle (.aab)  
eficiente para Play  
Store.

## **flutter build ipa**

Crea un paquete  
iOS (.ipa) listo para  
subir a App Store  
Connect.

## **flutter build web**

Compila el proyecto  
web (HTML, JS, CSS)  
para desplegar en  
servidores.

# Herramientas de Automatización y CI/CD

Automatizan la construcción, prueba y despliegue de aplicaciones.



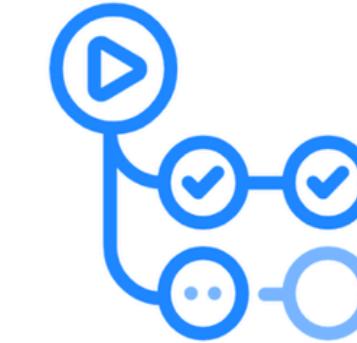
## Fastlane

Automatiza tareas repetitivas como capturas, metadatos y publicación.



## Codemagic

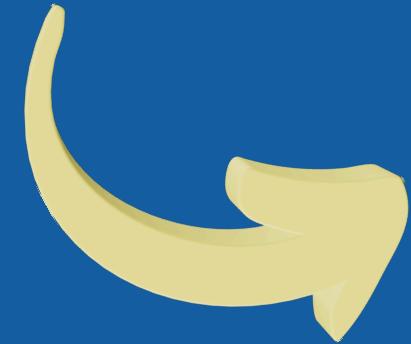
Servicio CI/CD para Flutter que gestiona pipelines automáticamente.



## GitHub Actions

Flujos de trabajo personalizados de CI/CD integrados en GitHub.

# HERRAMIENTAS DE GESTIÓN DE ENTORNOS Y CONFIGURACIÓN



**Gestionan variables de entorno (API keys, URLs) según el entorno (dev, staging, producción).**

- `lutter_dotenv`: Carga variables desde un archivo `.env` de forma segura.
- `flutter_config`: Similar, pero también expone variables a Android/iOS.

# Ejemplo de empaquetado en .apk

## 1.- Cambiar el icono de la app

### 1.1. instalar la dependencia

```
libreria_app> flutter pub add flutter_launcher_icons ---dev
```

### 1.2. Verificar la instalación

```
flutter_icons: ^0.0.0  
flutter_launcher_icons: ^0.14.4
```

### 1.3. Agregar el icono a las carpetas



### 1.4. Configurar el icono

```
flutter_launcher_icons:  
  android: true  
  ios: true  
  
  image_path: "assets/icon/icono.png"
```

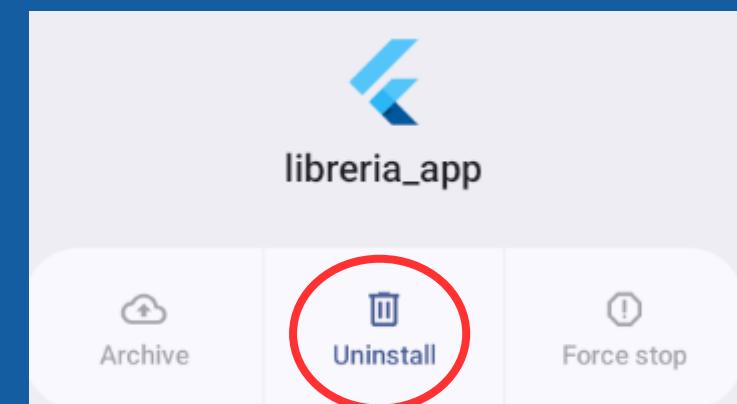
### 1.5. Ejecutar la dependencia

```
flutter pub run flutter_launcher_icons
```

### 1.6. Nos notifica que se sobrescribio el icono

- Creating default icons Android
- Overwriting the default Android launcher icon with a new icon

### 1.7. Desisntalar la app



### 1.8. Correr la app de nuevo

# Ejemplo de empaquetado en .apk

## 1.- Cambiar el icono de la app

### 1.9. Ya se nos habrá cambiado el icono



# Ejemplo de empaquetado en .apk

## 2.- Configurar la firma de la app

### 2.1. Crear el almacén de claves (keystore)

```
keytool -genkey -v -keystore Libreria.jks -keyalg RSA -keysize 2048 -validity 1000 -alias Libreria
```

### 2.2. Llenar los campos

Introduzca la contraseña del almacén de claves:

Volver a escribir la contraseña nueva:

Enter the distinguished name. Provide a single dot (.) to leave a sub-component empty or press ENTER to use the default value in braces.

¿Cuáles son su nombre y su apellido?

[Unknown]: Osbaldo Martinez

¿Cuál es el nombre de su unidad de organización?

[Unknown]: UTVM

¿Cuál es el nombre de su organización?

[Unknown]: UTVM

¿Cuál es el nombre de su ciudad o localidad?

[Unknown]: Ixmiquilpan

¿Cuál es el nombre de su estado o provincia?

[Unknown]: Hidalgo

¿Cuál es el código de país de dos letras de la unidad?

[Unknown]: MX

¿Es correcto CN=Osbaldo Martinez, OU=UTVM, O=UTVM, L=Ixmiquilpan, ST=Hidalgo, C=MX?

[no]: si

Generando par de claves RSA de 2,048 bits para certificado autofirmado (SHA384withRSA) con una validez de 1,000 días

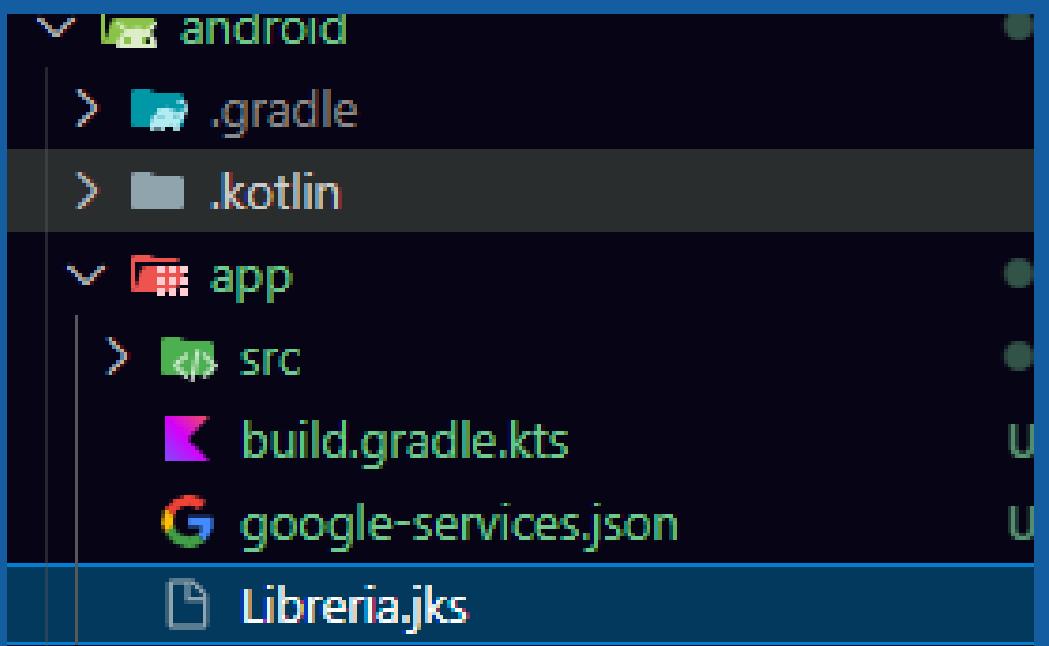
para: CN=Osbaldo Martinez, OU=UTVM, O=UTVM, L=Ixmiquilpan, ST=Hidalgo, C=MX

[Almacenando Libreria.jks]

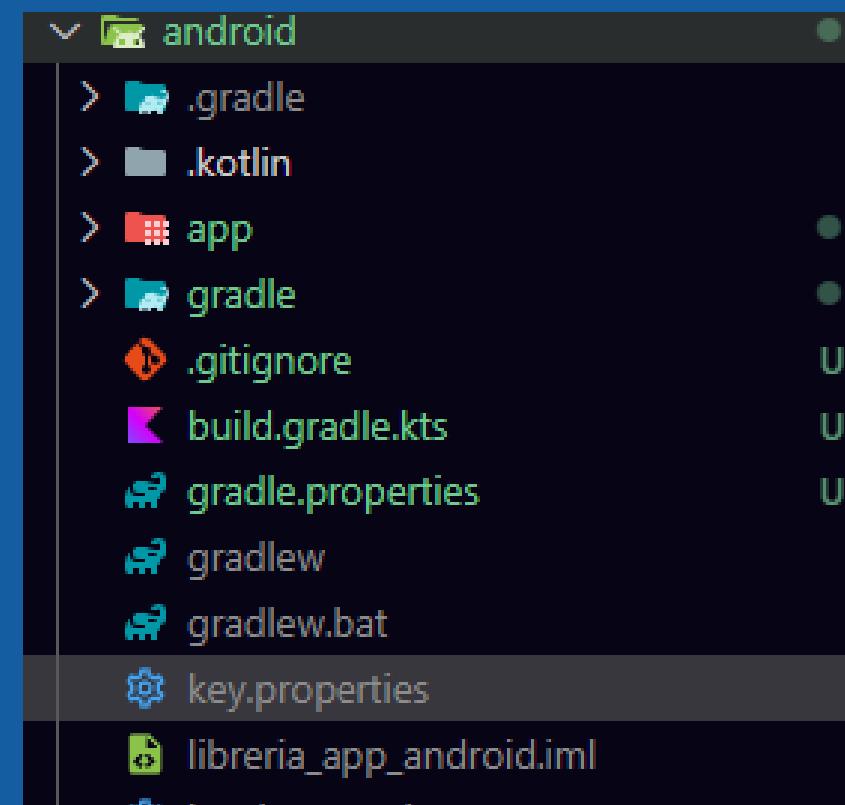
# Ejemplo de empaquetado en .apk

## 2.- Configurar la firma de la app

### 2.3. Mover el archivo generado



### 2.4. Crear “key.properties”



### 2.5. Escribir las contraseñas

```
#Contraseñas del Almacén de Claves
storePassword=wualo1506
keyPassword=wualo1506
keyAlias=Libreria
storeFile=Libreria.jks
```

# Ejemplo de empaquetado en .apk

## 3.- Compilar el APK

### 3.1. Limpiar el proyecto

```
flutter clean
```

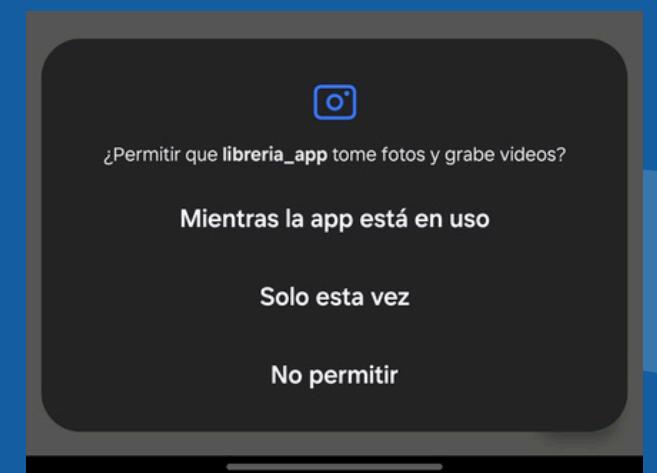
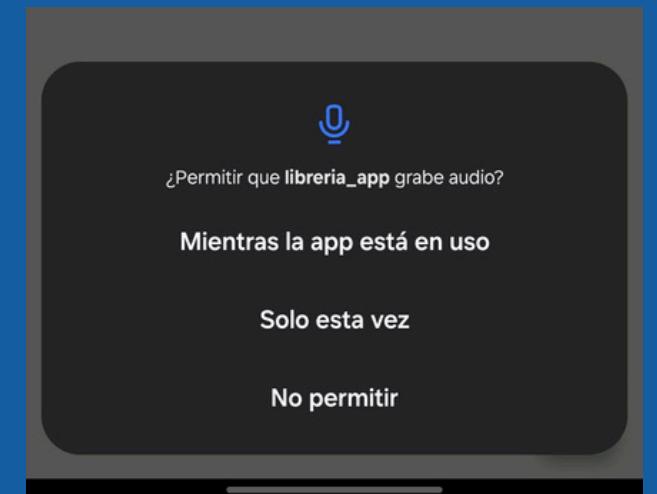
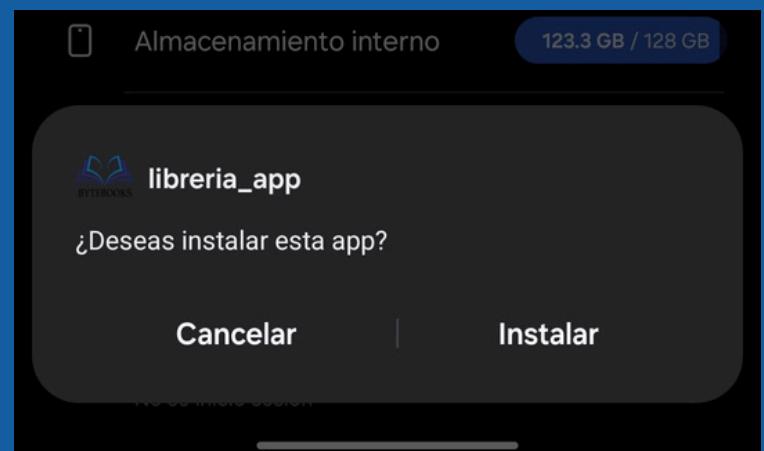
### 3.2. Comando para construir el apk

```
flutter build apk --release
```

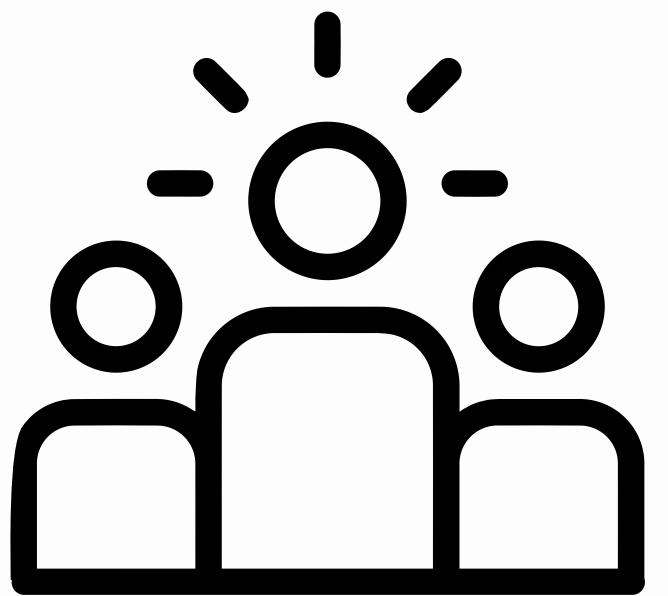
### 3.3. Ver la ruta donde se guardó

```
PS C:\Users\osbal\OneDrive\Documentos\respaldos\libreria_app> flutter build apk --release
Font asset "MaterialIcons-Regular.otf" was tree-shaken, reducing it from 1645184 to 1888 bytes (99.9% reduction). Tree-shaking can be disabled by providing the --no-tree-shake-icons flag when building your app.
Running Gradle task 'assembleRelease'...
117.2s
✓ Built build\app\outputs\flutter-apk\app-release.apk (48.6MB)
```

### 3.4. Instalar el APK y otorgar los permisos necesarios



# ¡GRACIAS!



# Resource Page

