Comandos Flutter - Guía Rápida

Índice

Configuración Inicial

Verificar instalación

```
flutter doctor
```

Muestra el estado de Flutter y todas las dependencias necesarias.

Actualizar Flutter

```
flutter upgrade
```

Actualiza Flutter a la última versión estable.

Establecer canal (channel)

```
flutter channel stable
flutter channel dev
flutter channel master
```

📡 ¿Qué es un Channel (Canal)?

Un **channel** es una rama de versiones de Flutter que puedes elegir. Cada canal tiene diferente nivel de estabilidad y actualización. Elige el que mejor se adapte a tus necesidades:

🟢 **STABLE (Recomendado - Para producción)**

```
```

```
flutter channel stable
```

- ✅ **Versiones probadas y confiables**
- ✅ Actualizaciones cada **3-4 meses**
- ✅ **Sin bugs críticos conocidos**
- ✅ Documentación completa
- ✅ Soporte oficial
- **Ideal para:** Proyectos en producción, cursos, empresas
- **Cambios:** Solo correcciones de seguridad entre versiones
- **Ejemplo:** v3.16, v3.19, v3.20

🟡 DEV (Intermedio - Para probar features)

```
flutter channel dev
```

- ⚠️ **Más nuevo que stable, pero más estable que master**
- ⚠️ Actualizaciones cada **1-2 semanas**
- ⚠️ Puede tener pequeños bugs
- ⚠️ Features en desarrollo
- ⚠️ Documentación parcial
- **Ideal para:** Probar features próximas, contribuidores
- **Cambios:** Features nuevas en desarrollo
- **Útil para:** Verificar compatibilidad antes de que salga en stable

🔴 MASTER (Experimental - Más nuevo pero inestable)

```
flutter channel master
```

- ❌ **El código más reciente y experimental**
- ❌ Actualizaciones **diariamente**
- ❌ Puede tener **bugs serios**
- ❌ Features sin terminar
- ❌ Documentación mínima
- ❌ **NO para producción**
- **Ideal para:** Desarrolladores internos de Flutter, investigación
- **Cambios:** Constantes y drásticos
- **Riesgo:** Muy alto de problemas

📊 Tabla Comparativa

Aspecto	Stable	Dev	Master
Estabilidad	☆☆☆☆☆	☆☆☆	☆

Aspecto	Stable	Dev	Master
Frecuencia actualizaciones	Trimestral	Semanal	Diaria
Features nuevas	☆	☆☆☆	☆☆☆☆☆
Para producción	✅ Sí	⚠ Caution	❌ No
Bugs esperados	Mínimos	Algunos	Muchos posibles
Documentación	Completa	Parcial	Mínima
Soporte oficial	✅ Sí	⚠ Limitado	❌ No

🎯 Recomendaciones por Situación

```
# Para tu CURSO (Getafe 2026) - USA ESTO
flutter channel stable

# Para probar features nuevas antes de lanzamiento
flutter channel dev

# Para contribuidores de Flutter (muy avanzado)
flutter channel master

# Cambiar entre canales
flutter upgrade                    # Actualiza al último del canal actual
flutter downgrade                  # Revierte a versión anterior
```

⚡ Cómo cambiar de canal

```
# Ver canal actual
flutter channel

# Ver todas las versiones disponibles en el canal actual
flutter version

# Cambiar a stable
flutter channel stable
flutter upgrade

# Cambiar a dev
flutter channel dev
flutter upgrade

# Cambiar a master
flutter channel master
flutter upgrade
```

⚠ Notas importantes

- ✅ **Recomendación:** Usa `stable` para todo (99% del tiempo)
- ⚠️ Al cambiar canal, ejecuta `flutter upgrade` para descargar la nueva versión
- ⚠️ Los proyectos creados en `master` pueden no funcionar en `stable`
- ⚠️ El tamaño de descarga puede variar (~600MB-2GB por actualización)
- 💾 Cada channel tiene su propia carpeta de caché en tu PC

Ver versión actual

```
flutter --version
```

Crear Proyectos

Crear proyecto nuevo

```
flutter create nombre_proyecto
```

Crear con diferentes templates

```
# Templates disponibles
flutter create -t app nombre_proyecto              # App completa (por defecto)
flutter create -t package nombre_paquete          # Paquete/librería reutilizable
flutter create -t plugin nombre_plugin            # Plugin nativo
flutter create -t module nombre_modulo            # Módulo Flutter
flutter create -t skeleton nombre_proyecto        # App con estructura básica
```

📊 Diferencias entre Templates

🔷 **APP (Por defecto)**

- **Uso:** Aplicación móvil/web lista para usar
- **Estructura:** Completa con ejemplo, assets, pubspec.yaml
- **Plataformas:** Android, iOS, Web, Desktop (todas)
- **Ideal para:** Tu primer proyecto, apps de producción
- **Ejemplo generado:** Counter app de demostración
- **Tamaño inicial:** ~15 MB con carpetas nativas
- **Cuándo usar:** 99% de los casos

📦 **PACKAGE**

- **Uso:** Librería/paquete reutilizable (sin UI)
- **Estructura:** Mínima, sin carpetas android/ios/web
- **Plataformas:** Dart puro (multiplataforma)
- **Ideal para:** Compartir código en pub.dev

- **Sin:** UI, assets, ejemplo visual
- **Tamaño inicial:** ~1 MB muy ligero
- **Cuándo usar:** Código reutilizable (validators, helpers, services)
- **Ejemplo:** `http`, `provider`, `dio`

🔌 PLUGIN

- **Uso:** Acceder a APIs nativas del SO
- **Estructura:** Incluye carpetas android/ e ios/
- **Lenguajes:** Dart + Kotlin + Swift
- **Ideal para:** Funciones nativas (cámara, GPS, sensores)
- **Requiere:** Conocimiento de Android/iOS nativo
- **Tamaño inicial:** ~20 MB con código nativo
- **Cuándo usar:** Necesitas acceder a hardware del teléfono
- **Ejemplo:** `camera`, `location`, `firebase_core`

📂 MODULE

- **Uso:** Módulo Flutter dentro de app nativa
- **Estructura:** Especial para integración
- **Ideal para:** Agregar Flutter a app nativa existente
- **Plataformas:** Específicamente Android y iOS
- **Tamaño inicial:** Similar a APP
- **Cuándo usar:** Tienes app nativa (Swift/Kotlin) y quieres Flutter en ella
- **Avanzado:** Para equipos grandes con múltiples plataformas

💀 SKELETON

- **Uso:** App básica con estructura mínima
- **Estructura:** APP pero sin código de ejemplo
- **Ideal para:** Empezar desde cero sin ejemplos
- **Plataformas:** Todas (como APP)
- **Tamaño inicial:** ~15 MB igual que APP
- **Diferencia vs APP:** Sin Counter app, más limpio
- **Cuándo usar:** Prefieres una hoja en blanco

Opciones avanzadas de creación

```
# Con lenguaje de programación específico
flutter create --platforms ios,android -t app nombre_proyecto
flutter create --platforms web -t app nombre_web

# Con organización específica
flutter create --org com.ejemplo nombre_proyecto

# Con descripción
flutter create --description "Mi aplicación" nombre_proyecto
```

```
# Crear sin ejemplos (más limpio)
flutter create --no-pub nombre_proyecto

# Crear en directorio específico
flutter create --offline .

# Combinadas
flutter create -t app \
  --org com.miempresa \
  --description "App de ecommerce" \
  mi_app
```

📝 Buenas Prácticas para Nombrar Proyectos

Reglas Generales

✅ Permitido:

- Letras minúsculas: `flutter create mi_app`
- Números (no al inicio): `flutter create app_v2`
- Guiones bajos: `flutter create mi_app_flutter`
- Guiones bajos y números combinados: `flutter create task_app_2`

❌ NO permitido:

- Mayúsculas: `flutter create MiApp` ❌
- Espacios: `flutter create mi app` ❌
- Guiones: `flutter create mi-app` ❌ (usa guiones bajos)
- Caracteres especiales: `flutter create mi@app` ❌
- Empezar con número: `flutter create 2app` ❌
- Palabras reservadas de Dart: `flutter create class`, `flutter create if`, `flutter create var` ❌

Convenciones Recomendadas

Tipo	Ejemplo	Descripción
Simple	`mi_app`	Nombre único y descriptivo
Específico	`task_manager_app`	Describe función principal
Con dominio	`ecommerce_platform`	Para proyectos corporativos
Versión	`app_v2`	Cuando hay múltiples versiones
Con prefijo	`flutter_mi_app`	Para librerías/packages públicos
Estudio	`flutter_curso_getafe`	Para proyectos educativos

Patrones por Contexto

Para Desarrollo Personal:

```
flutter create todo_app              # Claro y específico
flutter create contador_app          # Describe funcionalidad
flutter create weather_app           # Nombre simple
```

Para Proyectos Profesionales:

```
flutter create com.miempresa.payapp
flutter create --org com.miempresa mi_app
flutter create ecommerce_platform
flutter create crm_mobile
```

Para Cursos/Learning:

```
flutter create flutter_curso_basico
flutter create dart_learning_app
flutter create getafe_training_app
```

Para Librerías/Packages:

```
flutter create --org com.github mi_custom_package
flutter create flutter_custom_widgets
flutter create dart_utilities
```

Relación entre nombre del proyecto y organización (org)

```
# El nombre del proyecto y org juntos forman el package ID

# Estructura: org + nombre_proyecto
# Android: com.miempresa.mi_app
# iOS: com.miempresa.miApp (se convierte a camelCase)

# Ejemplo completo:
flutter create --org com.miempresa miapp
# Resultado: Android Package = com.miempresa.miapp

# Mejor práctica - org claro:
flutter create --org com.tu_empresa nombre_app
```

Ejemplos de Nombres Bien Estructurados

✅ **BIEN:**

```
flutter create notes_app
flutter create fitness_tracker
flutter create mobile_banking
flutter create social_network
flutter create task_manager
flutter create weather_forecast
flutter create gallery_app
flutter create chat_application
flutter create expense_tracker
flutter create meditation_app
```

✗ MAL:

```
flutter create NotesApp              # Mayúsculas
flutter create notes-app             # Guiones
flutter create notes app             # Espacio
flutter create notes@app             # Caracteres especiales
flutter create a                     # Muy corto, no descriptivo
flutter create mi_increible_aplicación_de_notas  # Muy largo
```

Cambiar Nombre Después de Creado

Si necesitas cambiar el nombre después:

```
# Opción 1: Crear nuevo con nombre correcto
flutter create nombre_correcto
# Y copiar el código

# Opción 2: Editar pubspec.yaml
# Cambiar nombre en la línea 1:
# name: nombre_viejo  →  name: nombre_nuevo

# Luego actualizar package IDs (más complejo)
```

Nombres para Diferentes Tipos de Proyectos

```
# App completa - Descriptivo
flutter create -t app biblioteca_municipal

# Package/Librería - Con prefijo flutter_
flutter create -t package flutter_custom_widgets

# Plugin nativo - Indica funcionalidad nativa
flutter create -t plugin camera_pro
```

```
# Módulo - Para apps nativas
flutter create -t module mi_modulo_flutter

# Skeleton - Nombre limpio
flutter create -t skeleton blank_app
```

Checklist Antes de Crear Proyecto

- ☐ Nombre en minúsculas
- ☐ Solo guiones bajos (sin guiones)
- ☐ Descriptivo y no muy largo (máx 30 caracteres)
- ☐ No es palabra reservada de Dart
- ☐ Organización (org) definida si es profesional
- ☐ Consistente con naming del equipo/empresa

Plataformas disponibles

```
# Especificar solo plataformas a soportar
flutter create -t app --platforms=ios,android,web mi_app
flutter create -t app --platforms=windows,macos,linux mi_app

# Plataformas individuales
flutter create -t app --platform=ios mi_app
flutter create -t app --platform=android mi_app
flutter create -t app --platform=web mi_app
flutter create -t app --platform=windows mi_app
flutter create -t app --platform=macos mi_app
flutter create -t app --platform=linux mi_app
```

Todas las opciones disponibles

```
# Ver todas las opciones
flutter create --help
```

Opciones principales:

- `-t, --template` - Template a usar (app, package, plugin, module, skeleton)
- `--platforms` - Plataformas a soportar (ios, android, web, windows, macos, linux)
- `--org` - Organización del paquete (ej: com.ejemplo)
- `-s, --sample` - ID de ejemplo de pub.dev a usar
- `--description` - Descripción del proyecto
- `--pub` - Ejecutar `flutter pub get` (por defecto true)
- `--offline` - Usar paquetes locales en caché
- `--verbose` - Mostrar output detallado

Ir al proyecto

```
cd nombre_proyecto
```

Desarrollo

Ejecutar app

```
flutter run
```

Compila y ejecuta en emulador/dispositivo conectado.

Ejecutar en específico

```
flutter run -d <device_id>        # En un dispositivo específico
flutter run -d chrome             # En navegador (web)
flutter run --debug               # Debug (por defecto, con hot reload)
flutter run --release             # Release optimizado sin debug (app final)
flutter run --profile             # Profile para analizar rendimiento
```

📊 Diferencias entre modos

Modo	Hot Reload	Performance	Tamaño	Uso
Debug	✅ Sí	Lenta	Grande	Desarrollo
Release	❌ No	Rápida ⚡	Pequeño	Producción final
Profile	❌ No	Normal	Normal	Profiling/análisis

- **Debug:** Incluye información de debug, permite hot reload, más lenta
- **Release:** Versión optimizada, sin debug, 10x más rápida
- **Profile:** Entre debug y release, para analizar performance sin overhead de debug

Hot Reload (durante ejecución)

```
r          # Hot reload (mantiene estado)
R          # Hot restart (reinicia la app)
q          # Salir
```

Ver dispositivos disponibles

```
flutter devices
```

Logs en tiempo real

```
flutter logs
```

Iniciar emulador

```
flutter emulators
flutter emulators launch <emulator_id>
```

Build y Compilación

Build Android

APK (para testing)

```
flutter build apk
```

Crea un APK en `build/app/outputs/flutter-apk/app-release.apk`

Bundle (para Google Play Store)

```
flutter build appbundle
```

Crea un bundle en `build/app/outputs/bundle/release/app-release.aab`

APK en debug

```
flutter build apk --debug
```

Build iOS

IPA (para App Store)

```
flutter build ios
```

Release

```
flutter build ios --release
```

Build Web

Producción

```
flutter build web
```

Desarrollo

```
flutter build web --dev
```

Build Desktop

Windows

```
flutter build windows
```

macOS

```
flutter build macos
```

Linux

```
flutter build linux
```

Dependencias

Agregar dependencia

```
flutter pub add nombre_paquete
```

Agregar versión específica

```
flutter pub add nombre_paquete:^1.0.0
```

Obtener/actualizar dependencias

```
flutter pub get
```

Actualizar dependencias

```
flutter pub upgrade
```

Actualizar a última versión

```
flutter pub upgrade --major-versions
```

Ver dependencias desactualizadas

```
flutter pub outdated
```

Ver información de paquete

```
flutter pub show nombre_paquete
```

Publicar paquete (en pub.dev)

```
flutter pub publish
```

Testing

Ejecutar tests unitarios

```
flutter test
```

Test específico

```
flutter test test/widget_test.dart
```

Tests con cobertura

```
flutter test --coverage
```

Tests de integración

```
flutter drive --target=test_driver/app.dart
```

Verificar análisis de código

```
flutter analyze
```

Formato de código

```
dart format lib/
flutter format .
```

Limpieza

Limpiar compilaciones previas

```
flutter clean
```

Eliminar archivo pubspec.lock y reinstalar

```
flutter clean
flutter pub get
```

Información del Sistema

Información detallada

```
flutter doctor -v
```

Versión verbose con detalles completos.

Ver configuración

```
flutter config
```

Localización de Android SDK

```
flutter config --android-sdk <ruta>
```

Ver todas las opciones disponibles

```
flutter --help
flutter <comando> --help
```

Shortcuts Útiles en Terminal

Tecla	Acción
r	Hot reload
R	Hot restart
h	Mostrar ayuda
d	Desconectar
w	Abrir DevTools
i	Inspeccionar/alternar modo inspector
p	Alternar widget tree
q	Salir

Comandos Avanzados

Generar APK con versión customizada

```
flutter build apk --build-number=2
```

Usar flavor específico

```
flutter run --flavor production -t lib/main_production.dart
```

Ver tamaño de app

```
flutter build apk --analyze-size
```

Verbose output (debug)

```
flutter run -v
```

Build con modo profiler

```
flutter build apk --profile
```

Workflow Típico de Desarrollo

```
# 1. Crear proyecto
flutter create mi_app
cd mi_app

# 2. Verificar ambiente
flutter doctor

# 3. Agregar dependencias
flutter pub add provider
flutter pub add http

# 4. Desarrollo
flutter run

# 5. Durante desarrollo (en terminal)
# Presiona 'r' para hot reload

# 6. Testing
flutter test

# 7. Análisis
flutter analyze

# 8. Formato
```

```
flutter format .

# 9. Build final
flutter build apk              # Android
# o
flutter build ios              # iOS
```

Tips Importantes

✅ **Siempre ejecutar** `flutter doctor` antes de empezar un nuevo proyecto

✅ **Usar** `flutter clean` si tienes problemas extraños

✅ **Los cambios en pubspec.yaml** requieren `flutter pub get` o hot restart

✅ **Hot reload NO funciona** con cambios en `main()` o `initState()`

✅ **Usar** `flutter analyze` regularmente para detectar problemas

✅ **Documentación oficial**: https://flutter.dev/docs/reference/flutter-cli

---

Opciones de `flutter run`

Con diferentes configuraciones

```
flutter run                          # Debug (por defecto)
flutter run --release               # Release optimizado
flutter run --profile               # Profile (análisis performance)
flutter run --debug                 # Debug explícito

# En dispositivo específico
flutter run -d <device_id>
flutter run -d all                  # En todos los dispositivos

# En navegador
flutter run -d chrome
flutter run -d firefox
flutter run -d edge
flutter run -d safari               # Solo macOS

# Opciones de debugging
flutter run --verbose               # Output detallado
flutter run --start-paused          # Pausado al inicio
flutter run --use-test-fonts        # Fuentes de test
flutter run --observatory-port=<puerto>  # Puerto específico
```

Opciones de `flutter build`

Construcción Android avanzada

```
flutter build apk --split-per-abi              # APK por arquitectura
flutter build apk --target-platform android-arm
flutter build apk --target-platform android-arm64
flutter build apk --target-platform android-x86
flutter build apk --target-platform android-x86_64

flutter build appbundle --verbose             # Con output detallado
flutter build apk --analyze-size               # Analizar tamaño
flutter build apk --build-number=5             # Versión custom
flutter build apk --build-name=1.0.0           # Nombre de versión
```

Construcción iOS avanzada

```
flutter build ios --release                    # Release para App Store
flutter build ios --no-codesign               # Sin firma de código
flutter build ipa                              # IPA para distribución
flutter build ios --verbose                    # Output detallado
```

Web avanzada

```
flutter build web --web-renderer html          # Renderer HTML
flutter build web --web-renderer canvaskit     # Renderer CanvasKit
flutter build web --csp                        # Content Security Policy
flutter build web --split-debuginfo            # Separar info debug
```

Opciones de Testing avanzadas

```
flutter test --verbose                         # Output detallado
flutter test --coverage                        # Con cobertura
flutter test --update-goldens                  # Actualizar imágenes de referencia
flutter test --tags=integration               # Solo tests con tag "integration"
flutter test --exclude-tags=slow               # Excluir tests lentos
flutter test test/models/                      # Solo carpeta específica
flutter test --concurrency=4                   # Parallelizar tests
flutter test --reporter=json                   # Output en JSON
```

Opciones de Pub (Dependencias)

```
flutter pub get                    # Obtener dependencias
flutter pub get --offline          # Solo paquetes en caché
flutter pub upgrade                # Actualizar todo
flutter pub upgrade --dry-run       # Ver cambios sin aplicar

flutter pub remove nombre_paquete   # Remover paquete
flutter pub cache repair            # Reparar caché local
flutter pub downgrade              # Downgrade a versiones anteriores

flutter pub outdated --mode=null-safety   # Ver si hay problemas
```

DevTools (Herramienta de Debugging)

```
flutter pub global activate devtools

# Durante flutter run
devtools                           # Inicia DevTools
# o presiona 'w' en terminal durante flutter run

# DevTools URL típica: http://localhost:9100
```

Actualizado: Enero 2026 - Curso Flutter Getafe