



Flutter Cheat Sheet – Instalación y Comandos

Referencia rápida con los pasos, estructura inicial de proyectos y comandos más útiles de Flutter. Esta guía está pensada para ayudarte a arrancar proyectos de forma práctica y sin perder tiempo.



Estructura básica de una app Flutter

```
import 'package:flutter/material.dart';

void main() {
  runApp(MyApp()); // Punto de entrada de la app
}

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Mi primera app Flutter',
      home: HomePage(),
    );
  }
}

class HomePage extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(title: Text('Inicio')),
      body: Center(child: Text('¡Hola Mundo!')),
    );
  }
}
```

Analogía:

- `main.dart` → la puerta de entrada de la casa.
- `MaterialApp` → el interiorista que define el estilo, rutas y colores.
- `Scaffold` → el andamio: paredes, techo y estructura de cada pantalla.
- `Widgets` (`Text`, `Button`, etc.) → los muebles que llenan la casa.
- Estructura tipo muñecas rusas: cada widget contiene a otro hasta llegar al más pequeño.



Comandos principales de Flutter

```
// Crear un nuevo proyecto
flutter create mi_app

// Ejecutar la app
flutter run
flutter run -d chrome      # navegador Chrome
flutter run -d ios         # simulador iOS
flutter run -d emulator-5554 # emulador Android específico

// Generar binarios para distribución
flutter build apk          # APK Android
flutter build appbundle    # AAB Play Store
flutter build ios          # IPA App Store
flutter build web          # versión web
flutter build windows      # ejecutable Windows
flutter build linux        # ejecutable Linux
flutter build macos        # ejecutable macOS

// Gestión de dependencias
flutter pub get            # descarga las librerías del pubspec.yaml
flutter pub add http       # añade http al pubspec.yaml automáticamente
flutter pub add provider   # añade provider

// Ejecutar tests unitarios
flutter test
```

Atajo: Durante `flutter run`, puedes usar:

- `r` → Hot Reload

- R → Hot Restart

Gestión de Paquetes (pub.dev)

```
# pubspec.yaml
name: mi_app
description: Mi primera aplicación en Flutter
version: 1.0.0+1

environment:
  sdk: ">=3.0.0 <4.0.0"

dependencies:
  flutter:
    sdk: flutter
  http: ^0.13.6
  provider: ^6.1.1

dev_dependencies:
  flutter_test:
    sdk: flutter

// Instalar dependencias tras editar el archivo
flutter pub get
```

Nota: Todos los paquetes oficiales y de terceros están disponibles en pub.dev. Ejemplo: http para peticiones REST, provider para gestión de estado.

Hot Reload vs Hot Restart

```
// Hot Reload (r)
✓ Recarga solo los cambios realizados en el código.
✓ Mantiene el estado actual de la aplicación.
```

- ✓ Ideal para ajustes visuales como estilos, colores o textos.

```
// Hot Restart (R)
```

- ✓ Reinicia completamente la aplicación.

- ✓ Pierde el estado (reinicia variables y controladores).

- ✓ Útil cuando modificas inicializaciones globales.

Dart DevTools

```
// Instalar y activar DevTools
```

```
flutter pub global activate devtools
```

```
// Ejecutar la app con puerto de observación
```

```
flutter run --observatory-port=9200
```

```
// Abrir DevTools en el navegador
```

```
devtools
```

Incluye: Inspector de widgets en vivo, análisis de performance, monitoreo de memoria y consola de logs.

Extensiones recomendadas

```
// Visual Studio Code
```

- ✓ Flutter (autocompletado, snippets, debugging)

- ✓ Dart (soporte de lenguaje)

- ✓ Flutter Widget Snippets (atajos de widgets comunes)

- ✓ Pubspec Assist (gestión rápida de dependencias)

```
// Android Studio
```

- ✓ Flutter plugin (emuladores + debugging)

- ✓ Dart plugin (soporte de lenguaje)
- ✓ Flutter Enhancement Suite (productividad extra)

Despliegue en distintas plataformas

```
// Android
flutter build apk
flutter build appbundle

// iOS
flutter build ipa

// Web
flutter build web --web-renderer canvaskit

// Escritorio
flutter build windows
flutter build macos
flutter build linux
```

Consejo: Para publicar en Play Store o App Store, debes configurar firma y cuentas de desarrollador. En escritorio puedes distribuir directamente los ejecutables.

✓ Consejo final: Tras instalar Flutter, ejecuta siempre `flutter doctor` para verificar tu entorno. Recuerda que `flutter pub add <paquete>` es la forma más rápida y segura de integrar nuevas librerías en tu proyecto.