



Cheatsheet – Introducción a Flutter



Unidad 04 - Autor: Sergi García Barea

 Concepto	 Explicación Útil y Aplicación
 ¿Qué es Flutter?	Framework de Google para crear apps nativas multiplataforma (Android, iOS, Web, Desktop) desde una sola base de código. Usa UI declarativa, Hot Reload y tiene un motor gráfico propio.
 ¿Por qué usar Dart?	Lenguaje rápido y moderno que permite Hot Reload y compila a código nativo o JavaScript. Ideal para interfaces reactivas.
 Ventajas clave	<ul style="list-style-type: none">• Código único para todas las plataformas.• UI personalizada y consistente.• Comunidad activa y herramientas avanzadas.• Alto rendimiento sin puente JS.
 Widgets esenciales	<ul style="list-style-type: none">• MaterialApp: raíz de la app.• Scaffold: estructura de cada pantalla (AppBar, Body, Drawer, FAB).• Todo es un widget (como muñecas rusas: se anidan).
 Comandos útiles	<ul style="list-style-type: none">• flutter create mi_app → crear proyecto.• flutter run -d chrome → ejecutar en navegador.• flutter build apk/web → generar binarios.• flutter pub get/add → gestionar dependencias.
 pubspec.yaml	Archivo central donde defines: <ul style="list-style-type: none">• Nombre, versión, assets e iconos.• Paquetes como http, provider, flutter_localizations.
 Hot Reload vs Hot Restart	<ul style="list-style-type: none">• Hot Reload (r): cambios visuales rápidos, conserva el estado.• Hot Restart (R): reinicia toda la app, útil para cambios estructurales.
 Testing básico	Crea archivos *_test.dart en /test. Usa flutter test para validarlos. Ejemplo: expect(sumar(2, 3), 5);
 Herramientas clave	<ul style="list-style-type: none">• Dart DevTools: inspección, rendimiento, memoria.• VS Code Extensions: Flutter, Dart, Snippets, Puspec Assist.
 Despliegue multiplataforma	Usa flutter build + plataforma (apk, web, windows, etc.). <ul style="list-style-type: none">• Web: usa canvaskit para mejor renderizado.• Android/iOS: firmar y subir a tiendas.• Desktop: compilar para distribución directa.