



Unidad 03 - Autor: Sergi García Barea

## 🚀 ¿Qué es Dart y por qué usarlo?

- **Lenguaje de programación moderno y optimizado**, creado por Google.
- Principalmente usado con **Flutter** para desarrollar apps móviles, web y escritorio con una sola base de código.
- **Compilación dual:**
  - **JIT (Just-In-Time)**: útil en desarrollo, permite Hot Reload (ver cambios al instante).
  - **AOT (Ahead-Of-Time)**: útil en producción, mejora el rendimiento.

## 📌 Variables y Tipado

### Declaración:

```
var nombre = 'Juan'; // tipo inferido automáticamente
String apellido = 'Pérez';
```

### Constantes:

```
final edad = 30;      // se asigna en tiempo de ejecución
const pi = 3.14;      // se conoce en tiempo de compilación
```

### Null safety:

```
String? nombre;        // puede ser nulo
nombre = 'Carlos';

late String apellido; // inicializada más tarde, pero no nula
apellido = 'Sánchez';
```

## 🔄 Control de Flujo

### Condicionales:

```
if (edad >= 18) {
  print('Mayor de edad');
} else {
  print('Menor de edad');
}
```



# Cheatsheet – Introducción a Dart



Unidad 03 - Autor: Sergi García Barea

## Switch con patrones:

```
switch (categoria) {  
  case 'A':  
    print('Categoría A');  
    break;  
  default:  
    print('Otra categoría');  
}
```

## Bucles:

```
for (var i = 0; i < 5; i++) print(i);  
  
while (condicion) { ... }  
  
do {  
  ...  
} while (condicion);
```

## Funciones

### Básicas:

```
int sumar(int a, int b) {  
  return a + b;  
}
```

### Arrow function:

```
int cuadrado(int x) => x * x;
```

### Parámetros:

```
String saludar(String nombre, {String? apellido = ''}) {  
  return 'Hola $nombre $apellido';  
}
```



Unidad 03 - Autor: Sergi García Barea

## Colecciones en Dart

### List:

```
var lista = [1, 2, 3];
lista.add(4);
```

### Set:

```
var conjunto = {1, 2, 3};
conjunto.add(3); // no se repite
```

### Map:

```
var persona = {'nombre': 'Ana', 'edad': 25};
print(persona['nombre']);
```

---

## Programación Orientada a Objetos

### Clase y constructor:

```
class Persona {
  String nombre;
  int edad;

  Persona(this.nombre, this.edad);
}
```

### Herencia y abstractas:

```
abstract class Animal {
  void hacerSonido();
}

class Perro extends Animal {
  @override
  void hacerSonido() => print('Guau');
}
```



# Cheatsheet – Introducción a Dart



Unidad 03 - Autor: Sergi García Barea

## Getters y setters:

```
class Producto {  
  double precio;  
  Producto(this.precio);  
  
  double get precioConIVA => precio * 1.21;  
}
```

## Extensiones:

```
extension Reversa on String {  
  String alReves() => split('').reversed.join();  
}  
  
print('hola'.alReves()); // aloh
```

## Callable Classes:

```
class Multiplicador {  
  int call(int x) => x * 2;  
}  
final m = Multiplicador();  
print(m(5)); // 10
```

# ⌚ Asincronía y Concurrencia

## Futures:

```
Future<String> obtenerDatos() async {  
  await Future.delayed(Duration(seconds: 2));  
  return 'Datos cargados';  
}
```

## Streams:

```
Stream<int> contador() async* {  
  for (int i = 0; i < 3; i++) {  
    yield i;  
    await Future.delayed(Duration(seconds: 1));  
  }  
}
```