Módulo 2: Fundamentos de Dart

¿Qué es Dart?

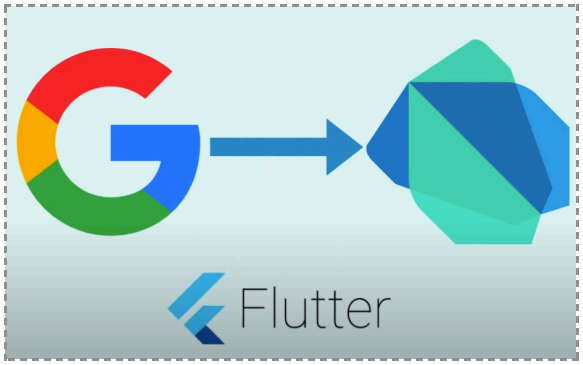
Dart es un lenguaje de programación moderno, desarrollado por Google, enfocado en la productividad y el rendimiento. Es el lenguaje principal para crear apps con Flutter, pero también se usa en web, backend y más.

Características principales de Dart

- Sintaxis clara y familiar (similar a JavaScript, Java, C#)
- Tipado estático y dinámico
- Soporte para programación orientada a objetos y funcional
- Hot reload (con Flutter)
- Gran rendimiento y multiplataforma

- Se dió a conocer en 2011.
- Se describe como una alternativa a Javascript.
- La compañía que está detrás es Google.
- Es Dart todo es un objeto.
- En el año 2013 se presenta la versión estable.
- Hoy en día tenemos la versión 2.8.4

Dart 3.10.7 • Flutter 3.38.7



Estructurado, Orientado a Objetos, Funcional, Asincrónico

Principales características

- AOT (Ahead Of Time): Compilado a un rápido y predecible código nativo. (Totalmente personalizable)
- Puede ser JIT (Just In Time): Compilado para una velocidad excepcional de desarrollo. (Esto incluye el popular Hot Reload).
- Hace fácil la creación de animaciones y transiciones que corren a 60fps (frames per second).
- Al ser compilado a código nativo, no hay puentes innecesarios para correr el código.
- Dart le permite a Flutter evitar el desarrollo de diseños en archivos independientes como JSX, XML o bien interfaces separadas.

Dart: Compilando

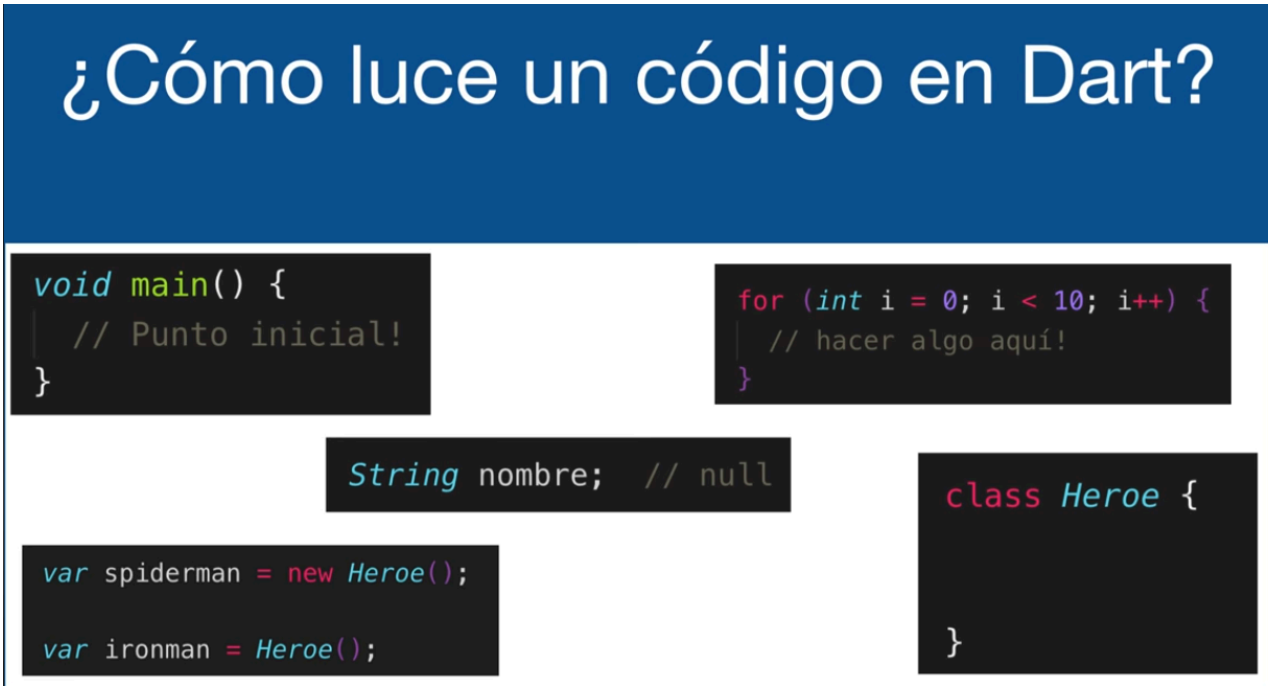
- ☐ JIT (just-in-time), durante el desarrollo:

  Hot reload / hot restart

- ☐ AOT (ahead-of-time), al compilar para producción:

  Código nativo, ARM y x64

1 Sintaxis básica de Dart



2 Variables, tipos de datos y operadores

```dart
void main() {
  int edad = 25;
  double altura = 1.75;
  String nombre = 'Ana';
  bool esEstudiante = true;
  var ciudad = 'Quito'; // Inferencia de tipo

  print('Nombre: $nombre, Edad: $edad');
}
```

Operadores aritméticos

Dart tiene compatibilidad con los operadores aritméticos más conocidos:

```dart
// + (sumar)
// - (restar)
// * (multiplicar)
// / (dividir)
// % (módulo)
```

Igualdad y operadores relacionales

Veremos los operadores de igualdad y relacionales más usados.

```
// == (igual)
// != (diferente)
// > (mayor que)
// < (menor que)
// >= (mayor igual que)
// <= (menor igual que)
```

[link referencia]

3 Funciones y parámetros

Las funciones en Dart son bloques de código reutilizables que permiten ejecutar una serie de instrucciones. Pueden aceptar parámetros y devolver un valor. Las funciones son fundamentales para estructurar y organizar el código.

```
int valor1=50, valor2=30;

int sumar(){
  return valor1 + valor2;
}
String DimeNombre(){
  return 'informaticonfig';
}

void main(){
  print(sumar());
  print(DimeNombre());
}
```

```
void DimeNombre(){
  print('informaticonfig');
}
String DameNombre(){
  return 'Jose Feliciano';
}


void main(){
  DimeNombre();
  print(DameNombre());
}
```

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

void saludar(String nombre) => print('Hola, $nombre!');

void main() {
  print(sumar(2, 3));
  saludar('Ana');
}
```

No 1

Link

No 2

Funciones en Dart: Guía Definitiva para Declararlas y Utilizarlas

4 Estructuras de control (if, for, while, switch)

página oficial

```dart
void main(){
  print('Ingresa tu edad');
  int edad = int.parse(stdin.readLineSync()!);

  if(edad >= 18){
    print('eres mayor de edad');
  }
  else if(edad>=10){
    print('eres un adolescente');
  }
  else{
    print('eres un niño');

  }

  int numero = 11;
  String resutlado = (numero % 2==0) ? 'par' : 'impar';
  print(resutlado);



}
```

5 Conjuntos listas y mapas

6 Clases y objetos

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

  Persona(this.nombre, this.edad);

  void saludar() {
    print('Hola, soy $nombre y tengo $edad años.');
  }
}

void main() {
  var persona = Persona('Lucía', 28);
  persona.saludar();
}
```

```
1     class Persona{
2         String? nombre, apellido;
3         int? edad;
4
5
6
7        void DimeDatos(){
8          print('Nombre: $nombre $apellido');
9        }
10
11    }
12
13    void main(){
14      Persona persona1 = Persona();
15      persona1.nombre = 'Jose';
16      persona1.apellido='Feliciano';
17      persona1.DimeDatos();
18
19
20    }
```

```dart
class Auto{
  String? marca, modelo;
  int? anio;

  //getter
  String get DimeMarca => marca ?? 'No definida';

  //setter
 set LaMarca(String valor){
  if (valor.isNotEmpty){
    marca = valor;
  }else{
    print('La marca no debe estar vacia');
  }
 }

}
void main(){
  Auto auto1 = Auto();
  auto1.LaMarca = '';
  print(auto1.DimeMarca);

}
```

7 Herencia y polimorfismo

```dart
class Animal{
  String? nombre, raza;
  int? tamanio;

  Animal(this.nombre, this.raza, this.tamanio);

  void comer(){
    print('$nombre está comiendo');
  }
}

class Perro extends Animal{

   Perro(String nombre, String raza, int tamanio)
      :super(nombre, raza,tamanio);


   void comer(){
     print('$nombre está comiendo');
   }


}
```

```dart


void main(){
  Animal objeto1 = Animal('Garfield', 'Angora', 50);
  objeto1.comer();
  Perro objeto2 = Perro('Firulais', 'Chiwawa', 25);
  objeto2.comer();


}
```

```dart
class Animal {
  void hablar() => print('Hace un sonido');
}

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

void main() {
  Animal miPerro = Perro();
  miPerro.hablar(); // Guau!
}
```

8 Manejo de excepciones (try/catch)

```dart
void main(){
    print('Ingrese valor1...');
    int valor1 = int.parse(stdin.readLineSync()!);
    print('Ingrese valor2...');
    int valor2 = int.parse(stdin.readLineSync()!);

    try{
      int resultado = valor1 ~/ valor2;
      print('Resultado: $resultado');

    }catch(ERROR){
      print('Error en operacion: $ERROR');
    }



}
```

9 Futures y async/await (introducción)

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

void main() async {
  print('Cargando...');
  String datos = await obtenerDatos();
  print(datos);
}
```

Link de referencia

No 1

Learn Dart Programming Language

No 2
Introducción a Dart - Para Flutter

No 3
Curso de Dart-Flutter desde cero para principiantes hasta nivel avanzado 2025

No 4
Aprende DART desde CERO (El lenguaje de Flutter)

No 5
Curso Dart - OpenBootcamp