



# Guía de Laboratorio1

## Introducción a la Programación con DART

M. Sc. Huáscar Gonzales

15 de marzo de 2025

Dart es un lenguaje de programación desarrollado por Google, diseñado para ser eficiente, flexible y fácil de aprender. Es el lenguaje principal utilizado en el framework Flutter para el desarrollo de aplicaciones móviles multiplataforma. Dart combina características de lenguajes modernos como la tipificación estática opcional, la programación orientada a objetos y la asincronía, lo que lo hace ideal para construir aplicaciones rápidas y escalables. Además, su sintaxis clara y concisa permite a los desarrolladores escribir código mantenible y productivo, tanto en el frontend como en el backend.

## Objetivo

El objetivo de esta guía es familiarizarte con el lenguaje Dart, su sintaxis básica y su entorno de desarrollo. Al finalizar, serás capaz de escribir y ejecutar programas simples en Dart.

## Requisitos Previos

- Conocimientos básicos de programación (variables, funciones, estructuras de control).
- Una computadora con acceso a internet.
- Un editor de texto o un entorno de desarrollo integrado (IDE) como Visual Studio Code.

## Instalación y Configuración

### 1. Instalar Dart SDK

- Ve a la página oficial de Dart: <https://dart.dev/get-dart>.
- Sigue las instrucciones para instalar el SDK en tu sistema operativo (Windows, macOS o Linux).
- Verifica la instalación ejecutando en la terminal:

```
1 dart --version
```

Esto debería mostrar la versión de Dart instalada.

## 2. Configurar un IDE

- Recomendamos usar **Visual Studio Code** (VS Code).
- Instala la extensión oficial de Dart en VS Code:
  1. Abre VS Code.
  2. Ve a la pestaña de extensiones (Ctrl+Shift+X).
  3. Busca Dart y haz clic en **Instalar**.

## 3. Crear un Proyecto

- Abre una terminal y ejecuta:

```
1 dart create mi_primer_proyecto
```

- Esto creará una carpeta llamada `mi_primer_proyecto` con una estructura básica de proyecto.

```
mi_primer_proyecto/  
- bin/  
  - mi_primer_proyecto.dart # Punto de entrada del programa  
- lib/                       # Biblioteca de código  
- test/                     # Pruebas unitarias  
- pubspec.yaml              # Archivo de configuración del proyecto
```

## Ejercicios Prácticos

### 1. Hola Mundo

- Abre el archivo `bin/mi_primer_proyecto.dart`.
- Reemplaza el contenido con el siguiente código:

```
1 void main() {  
2   print('Hola, Mundo');  
3 }
```

- Ejecuta el programa desde la terminal:

```
1 dart run bin/mi_primer_proyecto.dart
```

- Deberías ver el mensaje `Hola, Mundo!` en la consola.

### 2. Variables y Tipos de Datos

- Modifica el archivo `bin/mi_primer_proyecto.dart` para declarar variables:

```
1 void main() {  
2     String nombre = 'Dart';  
3     int version = 3;  
4     double puntuacion = 9.5;  
5     bool esDivertido = true;  
6  
7     print('Lenguaje: $nombre');  
8     print('Versión: $version');  
9     print('Puntuación: $puntuacion');  
10    print('¿Es divertido? $esDivertido');  
11 }
```

- Ejecuta el programa y observa la salida.

### 3. Funciones

- Crea una función que sume dos números:

```
1 int sumar(int a, int b) {  
2     return a + b;  
3 }  
4  
5 void main() {  
6     int resultado = sumar(5, 3);  
7     print('El resultado es: $resultado');  
8 }
```

- Ejecuta el programa y verifica el resultado.

### 4. Estructuras de Control

- Prueba un bucle for y una condición if:

```
1 void main() {  
2     for (int i = 1; i <= 5; i++) {  
3         if (i % 2 == 0) {  
4             print('$i es par');  
5         } else {  
6             print('$i es impar');  
7         }  
8     }  
9 }
```

- Ejecuta el programa y observa la salida.

## Retos Adicionales

- **Listas y Mapas:** Crea una lista de nombres y un mapa que asocie cada nombre con su edad. Imprime los valores.



- **Clases y Objetos:** Define una clase `Persona` con propiedades como `nombre` y `edad`. Crea un objeto e imprime sus detalles.
- **Manejo de Excepciones:** Escribe un programa que divida dos números y maneje la excepción de división por cero.

## Recursos Adicionales

- Documentación oficial de Dart: <https://dart.dev/guides>.
- DartPad (para probar código en línea): <https://dartpad.dev/>.
- Tutoriales en YouTube.

## Conclusión

Con esta guía, has configurado tu entorno de desarrollo en Dart y has escrito tus primeros programas. ¡Sigue practicando y explorando las características avanzadas del lenguaje!