



Guía de Laboratorio 2

Listas, Mapas, Clases y Manejo de Excepciones en Dart

M. Sc. Huáscar Gonzales

20 de marzo de 2025

Objetivo

El objetivo de esta guía es profundizar en el uso de estructuras de datos avanzadas como listas y mapas, así como en la creación de clases y objetos en Dart. Además, se introducirá el manejo de excepciones para escribir programas más robustos.

Requisitos Previos

- Haber completado la Guía de Laboratorio 1: Introducción a Dart.
- Conocimientos básicos de programación orientada a objetos (POO).
- Un entorno de desarrollo configurado con Dart SDK y un IDE como Visual Studio Code.

Ejercicios Prácticos

1. Listas y Mapas

Las listas y mapas son estructuras de datos fundamentales en Dart. A continuación, se presentan ejemplos para trabajar con ellas.

1.1. Listas

- Crea una lista de nombres y recórrela para imprimir cada elemento:

```
1 void main() {  
2     List<String> nombres = ['Ana', 'Juan', 'Carlos', 'Luisa'];  
3  
4     for (String nombre in nombres) {  
5         print('Nombre: $nombre');  
6     }  
7 }
```

- Ejecuta el programa y observa la salida.

1.2. Mapas

- Crea un mapa que asocie nombres con edades y recórrelo para imprimir los valores:

```
1  void main() {  
2      Map<String, int> edades = {  
3          'Ana': 25,  
4          'Juan': 30,  
5          'Carlos': 22,  
6          'Luisa': 28  
7      };  
8  
9      edades.forEach((nombre, edad) {  
10         print('$nombre tiene $edad años');  
11     });  
12 }
```

- Ejecuta el programa y observa la salida.

2. Clases y Objetos

Dart es un lenguaje orientado a objetos. A continuación, se muestra cómo definir clases y crear objetos.

2.1. Definición de una Clase

- Define una clase `Persona` con propiedades y métodos:

```
1  class Persona {  
2      String nombre;  
3      int edad;  
4  
5      // Constructor  
6      Persona(this.nombre, this.edad);  
7  
8      // Método para mostrar detalles  
9      void mostrarDetalles() {  
10         print('Nombre: $nombre, Edad: $edad');  
11     }  
12 }
```

2.2. Creación de Objetos

- Crea objetos de la clase `Persona` y utiliza sus métodos:

```
1  void main() {  
2      Persona persona1 = Persona('Ana', 25);  
3      Persona persona2 = Persona('Juan', 30);  
4  
5      persona1.mostrarDetalles();  
6      persona2.mostrarDetalles();  
7  }
```

- Ejecuta el programa y observa la salida.

3. Manejo de Excepciones

El manejo de excepciones permite gestionar errores en tiempo de ejecución. A continuación, se muestra un ejemplo.

3.1. División con Manejo de Excepciones

- Escribe un programa que divida dos números y maneje la excepción de división por cero:

```
1  void main() {  
2      int a = 10;  
3      int b = 0;  
4  
5      try {  
6          double resultado = a / b;  
7          print('El resultado es: $resultado');  
8      } on IntegerDivisionByZeroException {  
9          print('Error: División por cero no permitida.');10     } catch (e) {  
11         print('Ocurrió un error: $e');12     } finally {  
13         print('Fin del programa.');14     }  
15 }
```

- Ejecuta el programa y observa cómo se maneja la excepción.

Retos Adicionales

- **Listas y Mapas:** Crea una lista de objetos `Persona` y ordénala por edad.
- **Clases y Objetos:** Añade un método a la clase `Persona` que calcule el año de nacimiento basado en la edad.
- **Manejo de Excepciones:** Escribe un programa que lea un número desde la entrada del usuario y maneje excepciones si el valor no es un número válido.

Recursos Adicionales

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

Conclusión

En esta guía, has aprendido a trabajar con listas, mapas, clases, objetos y manejo de excepciones en Dart. Estos conceptos son fundamentales para desarrollar aplicaciones más complejas y robustas. ¡Sigue practicando y explorando las capacidades de Dart!