

## Trabajo Práctico: Introducción a Dart

### Objetivo

El objetivo de este TP es familiarizarse con el lenguaje de programación Dart, entendiendo su sintaxis y características básicas. Se busca que los estudiantes puedan aplicar lo aprendido en ejercicios prácticos que podrían ser utilizados en pruebas técnicas.

### Módulo 1: Fundamentos de Dart

#### Teoría

##### 1. Sintaxis básica y variables:

- Dart es un lenguaje tipado estáticamente.
- Declaración de variables: var, int, double, String, bool.
- Uso de final y const para declarar variables inmutables.

##### 2. Tipos de datos básicos:

- Números: int y double.
- Cadenas de texto: String.
- Booleanos: bool.

#### Ejercicios

##### 1. Ejercicio 1: Declaración de Variables

- Declara tres variables: un entero, un número decimal y una cadena de texto. Asigna valores iniciales y muestra estos valores en la consola.

##### 2. Ejercicio 2: Operaciones con Variables

- Escribe un programa que solicite al usuario dos números enteros. Realiza operaciones de suma, resta, multiplicación y división entre estos números e imprime los resultados.

### Módulo 2: Control de Flujo

#### Teoría

##### 1. Condicionales:

- Uso de if, else if, y else.
- Operadores de comparación (==, !=, >, <, >=, <=).

##### 2. Bucles:

- for: Ideal para iteraciones con un número conocido de repeticiones.
- while y do-while: Útiles cuando las repeticiones dependen de una condición.

#### Ejercicios

## Desarrollo Móvil

### 1. Ejercicio 3: Números Pares o Impares

- Escribe un programa que solicite un número al usuario y determine si es par o impar utilizando una estructura if.

### 2. Ejercicio 4: Sumatoria de Números

- Crea un programa que pida al usuario un número y luego sume todos los números desde 1 hasta el número proporcionado utilizando un bucle for.

## Módulo 3: Funciones

### Teoría

#### 1. Definición de funciones:

- Funciones básicas.
- Parámetros posicionales y nombrados.
- Retorno de valores.

#### 2. Funciones lambda:

- Sintaxis simplificada para funciones de una sola expresión.

### Ejercicios

#### 1. Ejercicio 5: Función de Área de un Círculo

- Define una función que calcule el área de un círculo dada su radio. Usa la fórmula  $\pi \times \text{radio}^2$ . Pide al usuario el radio y muestra el área calculada.

#### 2. Ejercicio 6: Función para Determinar Palíndromos

- Crea una función que reciba una cadena y devuelva true si es un palíndromo (se lee igual de adelante hacia atrás) y false en caso contrario. Prueba la función con diferentes entradas.

## Módulo 4: Clases y Objetos

### Teoría

#### 1. Clases en Dart:

- Definición de clases y creación de objetos.
- Propiedades y métodos.
- Constructores.

#### 2. Encapsulación:

- Getters y setters.

### Ejercicios

#### 1. Ejercicio 7: Clase Persona

## Desarrollo Móvil

- Define una clase Persona con propiedades nombre y edad. Crea un constructor para inicializar estas propiedades. Añade un método mostrarDatos que imprima los datos de la persona.

### 2. Ejercicio 8: Clase CuentaBancaria

- Crea una clase CuentaBancaria que tenga propiedades como titular, saldo, y métodos como depositar y retirar. Asegúrate de que no se pueda retirar más de lo que hay en la cuenta.

## Módulo 5: Colecciones

### Teoría

#### 1. Listas y Mapas:

- Listas (List): Colecciones ordenadas de elementos.
- Mapas (Map): Colecciones de pares clave-valor.

#### 2. Manipulación de Colecciones:

- Métodos comunes (add, remove, forEach, map, etc.).

### Ejercicios

#### 1. Ejercicio 9: Lista de Tareas

- Crea una lista para almacenar tareas pendientes. Permite al usuario agregar nuevas tareas y marcar tareas como completadas.

#### 2. Ejercicio 10: Agenda de Contactos

- Desarrolla un programa que use un mapa para almacenar nombres de contactos y sus números de teléfono. Permite agregar, eliminar y buscar contactos en la agenda.

### Ejercicios Optativos

#### Ejercicio 11: Recorrer y Manipular Arreglos

**Objetivo:** Practicar la manipulación y recorrido de arreglos en Dart.

##### 1. Operaciones Básicas con Arreglos

- Crea una lista de números enteros [3, 6, 9, 12, 15].
- Recorre la lista e imprime cada número en la consola.
- Calcula la suma de todos los números de la lista e imprime el resultado.
- Encuentra el número mayor y el menor en la lista y muéstralos en la consola.

##### 2. Filtrar y Transformar Datos

## Desarrollo Móvil

- A partir de la lista anterior, crea una nueva lista que contenga solo los números pares.
- Usa la función map para crear una nueva lista que contenga el doble de cada número de la lista original.
- Muestra ambas listas resultantes.

## Ejercicio 12: Registros y Agrupación

**Objetivo:** Utilizar clases para modelar registros y manipularlos.

### 1. Clase Estudiante

- Define una clase Estudiante que tenga propiedades como nombre, edad, y calificaciones (lista de enteros).
- Incluye un método promedioCalificaciones que calcule y devuelva el promedio de las calificaciones del estudiante.

### 2. Agrupación y Manipulación de Registros

- Crea una lista de objetos Estudiante con al menos 5 estudiantes.
- Escribe una función que reciba la lista de estudiantes y devuelva una nueva lista con los nombres de aquellos que tienen un promedio de calificaciones mayor a 6.
- Imprime los nombres de los estudiantes que cumplen esta condición.

## Ejercicio 13: Recursividad

**Objetivo:** Aprender y aplicar recursividad en problemas comunes.

### 1. Factorial de un Número

- Escribe una función recursiva llamada factorial que calcule el factorial de un número entero positivo. El factorial de un número  $n$  es el producto de todos los números enteros desde 1 hasta  $n$  (por ejemplo,  $5! = 5 \times 4 \times 3 \times 2 \times 1 = 120$ ,  $5! = 5 \times 4! = 5 \times 4 \times 3 \times 2 \times 1 = 120$ ).
- Prueba tu función con diferentes valores de entrada y muestra los resultados.

### 2. Serie de Fibonacci

- Crea una función recursiva llamada fibonacci que devuelva el número  $n$ -ésimo en la serie de Fibonacci. La serie de Fibonacci se define como:
  - $F(0) = 0$ ,  $F(1) = 1$
  - $F(n) = F(n-1) + F(n-2)$  para  $n > 1$

## Desarrollo Móvil

- Utiliza la función para calcular y mostrar los primeros 10 números de la serie de Fibonacci.

### 3. **Búsqueda Binaria Recursiva**

- Implementa una función recursiva llamada `busquedaBinaria` que realice una búsqueda binaria en una lista ordenada de números enteros.
- La función debe recibir como parámetros la lista, el valor a buscar, el índice inicial y el índice final.
- Prueba tu función con diferentes listas y valores para verificar su correcto funcionamiento.