

Creación y ejecución de una Activity

Autores

Alexis Chimba, Javier Ramos, Ariel Reyes, Anthony Villarreal

Ingeniera Doris Chicaiza

Universidad de las Fuerzas Armadas - ESPE

@espe.edu.ec, sjramos@espe.edu.ec, alreyes2@espe.edu.ec, anvillarreal@espe.edu.ec.

Resumen

La presente tarea fue implementada utilizando Flutter con el lenguaje Dart en Android Studio. Se desarrolló una aplicación móvil que permite resolver cuatro problemas algorítmicos distintos, cada uno con su respectiva lógica encapsulada en funciones y validaciones adecuadas para entrada de datos.

El primer problema consistió en calcular el sueldo semanal de un empleado con base en su edad y antigüedad en la compañía XYZ. Se desarrolló una interfaz que solicita dichos datos, realiza la suma de la serie aritmética correspondiente y muestra el resultado de forma clara al usuario.

El segundo problema requería el procesamiento de 100 números naturales, identificando cuántos están por debajo de 15, por encima de 50, cuántos se encuentran entre 25 y 45, y cuál es el promedio. Se construyó una rutina de entrada masiva de datos, acompañada de validaciones para garantizar que los números fueran naturales.

El tercer ejercicio consistió en determinar si dos números dados son amigos. Se calculó la suma de los divisores propios de cada número y se compararon los resultados entre sí.

Finalmente, el cuarto problema implicó la evaluación de una serie matemática tipo Taylor: $s = 1 + x + (x^2/2!) + (x^3/3!) + \dots (x^n/n!)$ para valores dados de x y n . Se emplearon ciclos y funciones auxiliares para calcular potencias y factoriales, manteniendo la precisión y validación de los datos.

La aplicación fue probada con diferentes entradas y mostró resultados correctos en cada caso

Palabras clave—Flutter, Dart, Validación, Algoritmos

I. Introducción

El objetivo de este laboratorio es desarrollar una aplicación móvil utilizando Flutter y Dart que permita resolver una serie de problemas algorítmicos fundamentales, aplicando estructuras de control, validación de datos y lógica matemática. Este tipo de práctica es relevante porque fortalece las habilidades de programación orientada a objetos y el diseño de interfaces interactivas en entornos móviles, competencias esenciales en el desarrollo de software moderno.

Durante el experimento, se abordaron distintos problemas que involucran cálculos matemáticos y análisis de datos: el cálculo del sueldo semanal en base a edad y antigüedad, la clasificación de un conjunto de números naturales bajo distintos criterios, la verificación de si dos números son amigos, y la evaluación de una serie matemática basada en potencias y factoriales. Cada uno de estos ejercicios fue implementado de forma modular dentro de la aplicación, permitiendo al usuario ingresar datos, recibir retroalimentación inmediata y comprender la lógica detrás de cada solución..

II. Trabajos Relacionados

El desarrollo de aplicaciones móviles utilizando Flutter y Dart ha ganado popularidad debido a su capacidad para crear aplicaciones multiplataforma eficientes y de alto rendimiento. Investigaciones recientes destacan las ventajas de Flutter en la simplificación del proceso de desarrollo y en la mejora de la experiencia del usuario (Kumar & Singh, 2023).

En el ámbito educativo, la enseñanza de algoritmos y el pensamiento computacional son fundamentales para formar habilidades de resolución de problemas. Estudios han demostrado que la educación en algoritmos mejora significativamente las habilidades de programación y la autoeficacia de los estudiantes (Tuna & Tuna, 2020).

Respecto a los números amigos, la investigación matemática ha explorado profundamente estas propiedades numéricas, proporcionando una base teórica sólida para su comprensión y aplicación en problemas computacionales (Brown, 2009).

La implementación de series de Taylor en programación es esencial en diversos campos, incluyendo el aprendizaje automático y la ingeniería. La serie de Taylor se utiliza para aproximar funciones complejas, optimizar algoritmos y comprender el comportamiento de modelos, siendo una herramienta poderosa en el análisis numérico (Chatterjee, 2021).

III. Materiales y Métodos

Materiales y Equipos Utilizados.

- Software de Desarrollo Android Studio
- Framework Flutter
- Lenguaje de programación Dart

- Emulador Android

IV. Desarrollo

1. Primero se creo un proyecto de flutter
2. Una vez con el proyecto de flutter creado, generamos el directorio de pantallas en la carpeta lib.



3. Este Código (calculo_sueldo.dart) en Flutter calcula el sueldo semanal de un empleado según su edad y años de antigüedad en la empresa XYZ. El usuario ingresa ambos valores, y tras validarlos, se aplica la fórmula $35000 + \text{edad} + 100 * \text{suma}(1..\text{antigüedad})$. El resultado se muestra en pantalla, y se incluye navegación entre pantallas para facilitar el uso de la app.

```
1 import 'package:flutter/material.dart';
2
3
4 class CalculoSueldo extends StatefulWidget {
5   const CalculoSueldo({super.key});
6
7   @override
8   State<CalculoSueldo> createState() => _CalculoSueldoState();
9 }
10
11 class _CalculoSueldoState extends State<CalculoSueldo> {
12   final TextEditingController _edadController = TextEditingController();
13   final TextEditingController _antiguedadController = TextEditingController();
14
15   String _resultado = '';
16
17   void _calcularSueldo() {
18     final int? edad = int.tryParse(_edadController.text);
19     final int? antiguedad = int.tryParse(_antiguedadController.text);
20
21     if (edad == null || edad <= 0) {
22       setState(() {
23         _resultado = 'Por favor ingrese una edad válida (> 0).';
24       });
25       return;
26     }
27
28     if (antiguedad == null || antiguedad < 0) {
29       setState(() {
30         _resultado = 'Por favor ingrese una antigüedad válida (>= 0).';
31       });
32       return;
33     }
34
35     int sumaAntiguedad = (antiguedad * (antiguedad + 1)) ~/ 2;
36     int sueldo = 35000 + edad + (100 * sumaAntiguedad);
37
38     setState(() {
39       _resultado = 'El sueldo semanal es: \${sueldo.toString()}';
40     });
41   }
42 }
```

```

42
43 @override
44 Widget build(BuildContext context) {
45   return Scaffold(
46     appBar: AppBar(
47       title: const Text('Cálculo de Sueldo'),
48       backgroundColor: Colors.lightBlueAccent,
49     ),
50     body: Padding(
51       padding: const EdgeInsets.all(24.0),
52       child: SingleChildScrollView( // Por si el teclado cubre los botones
53         child: Column(
54           mainAxisAlignment: MainAxisAlignment.center,
55           children: [
56             const Text(
57               'Ingrese los datos del empleado:',
58               style: TextStyle(fontSize: 20, fontWeight: FontWeight.bold),
59             ),
60             const SizedBox(height: 20),
61             TextField(
62               controller: _edadController,
63               keyboardType: TextInputType.number,
64               decoration: const InputDecoration(
65                 labelText: 'Edad',
66                 border: OutlineInputBorder(),
67               ),
68             ),
69             const SizedBox(height: 20),
70             TextField(
71               controller: _antiguedadController,
72               keyboardType: TextInputType.number,
73               decoration: const InputDecoration(
74                 labelText: 'Años en la empresa',
75                 border: OutlineInputBorder(),
76               ),
77             ),
78             const SizedBox(height: 20),
79             ElevatedButton(
80               onPressed: _calcularSueldo,
81               child: const Text('Calcular sueldo'),
82             ),
83             const SizedBox(height: 20),
84             Text(
85               _resultado,
86               style: const TextStyle(fontSize: 18, fontWeight: FontWeight.bold),
87               textAlign: TextAlign.center,
88             ),
89             const SizedBox(height: 30),
90             Row(
91               mainAxisAlignment: MainAxisAlignment.spaceEvenly,
92               children: [
93                 ElevatedButton.icon(
94                   onPressed: () {
95                     Navigator.pushReplacementNamed(context, '/'); // Ir al nodo raíz
96                   },
97                   icon: const Icon(Icons.arrow_back),
98                   label: const Text('Atrás'),
99                 ),
100                ElevatedButton.icon(
101                  onPressed: () {
102                    Navigator.pushNamedAndRemoveUntil(context, '/', (route) => false); // Volver a inicio limpio
103                  },
104                  icon: const Icon(Icons.home),
105                  label: const Text('Inicio'),
106                ),
107                ElevatedButton.icon(
108                  onPressed: () {
109                    Navigator.pushReplacementNamed(context, '/ analisis_100'); // Ir a la siguiente pantalla
110                  },
111                  icon: const Icon(Icons.arrow_forward),
112                  label: const Text('Siguiente'),
113                ),
114              ],
115            ),
116          ],
117        ),
118      ),
119    );
120  };
121 }
122 }
123

```

4. Este Código (analisis_100_numeros.dart) Flutter crea una pantalla que permite ingresar hasta 100 números naturales separados por comas, y luego realiza un análisis simple: cuenta cuántos son menores que 15, mayores que 50, cuántos están entre 25 y 45, y calcula el promedio. También incluye controles para reiniciar la entrada, mostrar resultados y navegar entre pantallas de la app.

```
1 import 'package:flutter/material.dart';
2 import 'package:tarea_1_1/pantallas/calculo_2_numeros.dart';
3 import 'package:tarea_1_1/pantallas/calculo_sueldo.dart';
4 import 'package:tarea_1_1/pantallas/pagina_inicial.dart'; // Asegúrate de importar correctamente
5
6 class Analisis100Numeros extends StatefulWidget {
7   @override
8   State<Analisis100Numeros> createState() => _Analisis100NumerosState();
9 }
10
11 class _Analisis100NumerosState extends State<Analisis100Numeros> {
12   final TextEditingController _numeroController = TextEditingController();
13   final List<int> _numeros = [];
14
15   String _resultado = '';
16
17   void _agregarNumerosDesdeTexto() {
18     final texto = _numeroController.text;
19     final partes = texto.split(',');
20     final nuevosNumeros = <int>[];
21
22     for (var parte in partes) {
23       final numeroStr = parte.trim();
24       if (numeroStr.isEmpty) continue;
25
26       final numero = int.tryParse(numeroStr);
27       if (numero == null || numero < 0) {
28         setState(() {
29           _resultado = 'Todos los valores deben ser números naturales (0 o mayores).';
30         });
31         return;
32       }
33
34       nuevosNumeros.add(numero);
35     }
36
37     if (_numeros.length + nuevosNumeros.length > 100) {
38       setState(() {
39         _resultado = 'El total excede los 100 números permitidos.';
40       });
41       return;
42     }
43
44     setState(() {
45       _numeros.addAll(nuevosNumeros);
46       _numeroController.clear();
47       _resultado = 'Se agregaron ${nuevosNumeros.length} números. Total actual: ${_numeros.length}.';
48     });
49   }
50
51
52   void _calcularResultados() {
53     if (_numeros.length != 100) {
54       setState(() {
55         _resultado = 'Debes ingresar exactamente 100 números antes de calcular.';
56       });
57       return;
58     }
59
60     int menores15 = _numeros.where((n) => n < 15).length;
61     int mayores50 = _numeros.where((n) => n > 50).length;
62     int entre25y45 = _numeros.where((n) => n >= 25 && n <= 45).length;
63     double promedio = _numeros.reduce((a, b) => a + b) / 100;
```

```

65     setState(() {
66       _resultado = ''
67     Resultados:
68     - Menores que 15: $menores15
69     - Mayores que 50: $mayores50
70     - Entre 25 y 45: $entre25y45
71     - Promedio: ${promedio.toStringAsFixed(2)}
72     ''
73   });
74 }
75
76 void _reiniciar() {
77   setState(() {
78     _numeros.clear();
79     _resultado = '';
80     _numeroController.clear();
81   });
82 }
83
84 @override
85 Widget build(BuildContext context) {
86   return Scaffold(
87     appBar: AppBar(
88       title: const Text('Análisis de 100 Números'),
89       backgroundColor: Colors.lightBlueAccent,
90     ),
91     body: Padding(
92       padding: const EdgeInsets.all(24.0),
93       child: SingleChildScrollView(
94         child: Column(
95           children: [
96             Text(
97               'Ingrese número ${_numeros.length + 1} de 100:',
98               style: const TextStyle(fontSize: 18),
99             ),
100             const SizedBox(height: 10),
101             TextField(
102               controller: _numeroController,
103               keyboardType: TextInputType.number,
104               decoration: const InputDecoration(
105                 labelText: 'Número natural',
106                 border: OutlineInputBorder(),
107               ),
108             ),
109             const SizedBox(height: 10),
110             ElevatedButton(
111               onPressed: _agregarNumerosDesdeTexto,
112               child: const Text('Agregar número(s)'),
113             ),
114             const SizedBox(height: 20),
115             ElevatedButton(
116               onPressed: _calcularResultados,
117               child: const Text('Calcular resultados'),
118             ),
119             const SizedBox(height: 10),
120             ElevatedButton(
121               onPressed: _reiniciar,
122               child: const Text('Reiniciar todo'),
123               style: ElevatedButton.styleFrom(
124                 backgroundColor: Colors.redAccent,
125               ),
126             ),
127             const SizedBox(height: 20),
128             Text(
129               _resultado,
130               style: const TextStyle(fontSize: 16),
131             ),
132             const SizedBox(height: 30),
133             Row(
134               mainAxisAlignment: MainAxisAlignment.spaceEvenly,
135               children: [
136                 ElevatedButton.icon(
137                   onPressed: () {
138                     Navigator.pushReplacementNamed(context, '/calculo_sueldo'); // Ir al nodo raíz
139                   },
140                   icon: const Icon(Icons.arrow_back),
141                   label: const Text('Atrás'),
142                 ),
143                 ElevatedButton.icon(
144                   onPressed: () {
145                     Navigator.pushNamedAndRemoveUntil(context, '/', (route) => false); // Volver a inicio limpio
146                   },
147                   icon: const Icon(Icons.home),
148                   label: const Text('Inicio'),
149                 ),
150               ],
151             ),
152           ],
153         ),
154       ),
155     ),
156   );
157 }

```

```

149         ),
150         ElevatedButton.icon(
151             onPressed: () {
152                 Navigator.pushReplacementNamed(context, '/calculo_2_numeros'); // Ir a la siguiente pantalla
153             },
154             icon: const Icon(Icons.arrow_forward),
155             label: const Text('Siguiente'),
156         ),
157     ],
158 ),
159 ],
160 ),
161 ),
162 ),
163 );
164 }
165 }
166

```

5. Este Código(calculo_2_numeros.dart) Flutter crea una pantalla que permite al usuario ingresar dos números enteros positivos y verificar si son números amigos. Dos números son amigos si la suma de los divisores propios del primero es igual al segundo, y viceversa. La pantalla incluye validaciones, botones para calcular, reiniciar y navegar entre otras pantallas de la app.

```

1 import 'package:flutter/material.dart';
2 import 'package:tarea_1_1/pantallas/analisis_100_numeros.dart';
3 import 'package:tarea_1_1/pantallas/calculo_serie_taylor.dart';
4 import 'package:tarea_1_1/pantallas/pagina_inicial.dart';
5
6 class Calculo2Numeros extends StatefulWidget {
7   @override
8   State<Calculo2Numeros> createState() => _Calculo2NumerosState();
9 }
10
11 class _Calculo2NumerosState extends State<Calculo2Numeros> {
12   final TextEditingController _aController = TextEditingController();
13   final TextEditingController _bController = TextEditingController();
14   String _resultado = '';
15
16   int sumaDivisoresPropios(int numero) {
17     int suma = 0;
18     for (int i = 1; i < numero; i++) {
19       if (numero % i == 0) {
20         suma += i;
21       }
22     }
23     return suma;
24   }
25
26   void _verificarNumerosAmigos() {
27     final int? a = int.tryParse(_aController.text);
28     final int? b = int.tryParse(_bController.text);
29
30     if (a == null || b == null || a <= 0 || b <= 0) {
31       setState(() {
32         _resultado = 'Por favor ingrese dos números enteros positivos válidos.';
33       });
34       return;
35     }
36
37     int sumaA = sumaDivisoresPropios(a);
38     int sumaB = sumaDivisoresPropios(b);
39

```



```

39
40   if (sumaA == b && sumaB == a) {
41     setState(() {
42       _resultado =
43         'Los números $a y $b SON amigos.\nSuma de divisores de $a = $sumaA\nSuma de divisores de $b = $sumaB';
44     });
45   } else {
46     setState(() {
47       _resultado =
48         'Los números $a y $b NO son amigos.\nSuma de divisores de $a = $sumaA\nSuma de divisores de $b = $sumaB';
49     });
50   }
51 }
52
53 void _reiniciar() {
54   setState(() {
55     _aController.clear();
56     _bController.clear();
57     _resultado = '';
58   });
59 }
60
61 @override
62 Widget build(BuildContext context) {
63   return Scaffold(
64     appBar: AppBar(
65       title: const Text('Cálculo de Números Amigos'),
66       backgroundColor: Colors.lightBlueAccent,
67     ),
68     body: Padding(
69       padding: const EdgeInsets.all(24.0),
70       child: SingleChildScrollView(
71         child: Column(
72           children: [
73             const Text(
74               'Ingrese dos números enteros positivos:',
75               style: TextStyle(fontSize: 18),
76             ),
77             const SizedBox(height: 10),
78             TextField(
79               controller: _aController,
80               keyboardType: TextInputType.number,
81               decoration: const InputDecoration(
82                 labelText: 'Número A',
83                 border: OutlineInputBorder(),
84               ),
85             ),
86             const SizedBox(height: 10),
87             TextField(
88               controller: _bController,
89               keyboardType: TextInputType.number,
90               decoration: const InputDecoration(
91                 labelText: 'Número B',
92                 border: OutlineInputBorder(),
93               ),
94             ),
95             const SizedBox(height: 15),
96             ElevatedButton(
97               onPressed: _verificarNumerosAmigos,
98               child: const Text('Verificar'),
99             ),
100            const SizedBox(height: 10),
101            ElevatedButton(
102              onPressed: _reiniciar,
103              style: ElevatedButton.styleFrom(backgroundColor: Colors.redAccent),
104              child: const Text('Reiniciar'),
105            ),
106            const SizedBox(height: 20),
107            Text(
108              _resultado,
109              style: const TextStyle(fontSize: 16),
110            ),
111            const SizedBox(height: 30),
112            Row(
113              mainAxisAlignment: MainAxisAlignment.spaceEvenly,
114              children: [
115                ElevatedButton.icon(
116                  onPressed: () {
117                    Navigator.pushReplacementNamed(context, '/analisis_100'); // Ir al nodo raíz
118                  },
119                  icon: const Icon(Icons.arrow_back),
120                  label: const Text('Atrás'),
121                ),

```



```

123         onPressed: () {
124             Navigator.pushNamedAndRemoveUntil(context, '/', (route) => false); // Volver a inicio limpio
125         },
126         icon: const Icon(Icons.home),
127         label: const Text('Inicio'),
128     ),
129     ElevatedButton.icon(
130         onPressed: () {
131             Navigator.pushReplacementNamed(context, '/serie_taylor'); // Ir a la siguiente pantalla
132         },
133         icon: const Icon(Icons.arrow_forward),
134         label: const Text('Siguiente'),
135     ),
136 ],
137 ),
138 ],
139 ),
140 ),
141 ),
142 );
143 }
144 }
145

```

6. Este código (calculo_serie_taylor.dart) define una pantalla de Flutter para calcular la **serie de Taylor** de la función exponencial e , truncada hasta el término n , según la fórmula: $s = 1 + x + (x^2/2!) + (x^3/3!) + \dots (x^n/n!)$

```

1 import 'package:flutter/material.dart';
2 import 'package:tarea_1_1/pantallas/calculo_2_numeros.dart';
3 import 'dart:math';
4 import 'package:tarea_1_1/pantallas/pagina_inicial.dart';
5
6 class CalculoSerieTaylor extends StatefulWidget {
7     const CalculoSerieTaylor({super.key});
8
9     @override
10     State<CalculoSerieTaylor> createState() => _CalculoSerieTaylorState();
11 }
12
13 class _CalculoSerieTaylorState extends State<CalculoSerieTaylor> {
14     final TextEditingController _xController = TextEditingController();
15     final TextEditingController _nController = TextEditingController();
16     String _resultado = '';
17
18     double calcularFactorial(int numero) {
19         double factorial = 1;
20         for (int i = 1; i <= numero; i++) {
21             factorial *= i;
22         }
23         return factorial;
24     }
25
26     void calcularSerieTaylor() {
27         final double? x = double.tryParse(_xController.text);
28         final int? n = int.tryParse(_nController.text);
29
30         if (x == null || n == null || n < 0) {
31             setState(() {
32                 _resultado = 'Por favor ingrese valores válidos:\n- x puede ser cualquier número.\n- n debe ser entero positivo o cero.';
33             });
34             return;
35         }
36
37         double suma = 1; // primer término
38         for (int i = 1; i <= n; i++) {
39             suma += (pow(x, i) / calcularFactorial(i));
40         }
41
42         setState(() {
43             _resultado = 'Resultado de la serie para x=$x y n=$n es:\n$suma';
44         });
45     }
46

```

```

47 void reiniciar() {
48   setState(() {
49     _xController.clear();
50     _nController.clear();
51     _resultado = '';
52   });
53 }
54
55 @override
56 Widget build(BuildContext context) {
57   return Scaffold(
58     appBar: AppBar(
59       title: const Text('Cálculo de Serie de Taylor'),
60       backgroundColor: Colors.lightBlueAccent,
61     ),
62     body: Padding(
63       padding: const EdgeInsets.all(24.0),
64       child: SingleChildScrollView(
65         child: Column(
66           children: [
67             const Text(
68               'Ingrese los valores de x y n:',
69               style: TextStyle(fontSize: 18),
70             ),
71             const SizedBox(height: 10),
72             TextField(
73               controller: _xController,
74               keyboardType: TextInputType.number,
75               decoration: const InputDecoration(
76                 labelText: 'Valor de x',
77                 border: OutlineInputBorder(),
78               ),
79             ),
80             const SizedBox(height: 10),
81             TextField(
82               controller: _nController,
83               keyboardType: TextInputType.number,
84               decoration: const InputDecoration(
85                 labelText: 'Valor de n (entero ≥ 0)',
86                 border: OutlineInputBorder(),
87               ),
88             ),
89             const SizedBox(height: 15),
90             ElevatedButton(
91               onPressed: calcularSerieTaylor,
92               child: const Text('Calcular Serie'),
93             ),
94             const SizedBox(height: 10),
95             ElevatedButton(
96               onPressed: reiniciar,
97               style: ElevatedButton.styleFrom(backgroundColor: Colors.redAccent),
98               child: const Text('Reiniciar'),
99             ),
100            const SizedBox(height: 20),
101            Text(
102              _resultado,
103              style: const TextStyle(fontSize: 16),
104            ),
105            const SizedBox(height: 30),
106            Row(
107              mainAxisAlignment: MainAxisAlignment.spaceEvenly,
108              children: [
109                ElevatedButton.icon(
110                  onPressed: () {
111                    Navigator.pushReplacementNamed(context, '/calcula_2_numeros'); // Ir al nodo raíz
112                  },
113                  icon: const Icon(Icons.arrow_back),
114                  label: const Text('Atrás'),
115                ),
116                ElevatedButton.icon(
117                  onPressed: () {
118                    Navigator.pushNamedAndRemoveUntil(context, '/', (route) => false); // Volver a inicio limpio
119                  },
120                  icon: const Icon(Icons.home),
121                  label: const Text('Inicio'),
122                ),
123                ElevatedButton.icon(
124                  onPressed: () {
125                    Navigator.pushReplacementNamed(context, '/'); // Ir a la siguiente pantalla
126                  },
127                  icon: const Icon(Icons.arrow_forward),
128                  label: const Text('Siguiente'),
129                ),
130              ],
131            ),
132          ],
133        ),
134      ),

```

7. El Código (pagina_inicial.dart) crea una pantalla inicial en Flutter con un AppBar que contiene el título "AlexGames Studios" y un cuerpo con un mensaje de bienvenida. Además, presenta cuatro botones que permiten al usuario navegar a diferentes pantallas de la aplicación: "Cálculo de Sueldo", "Análisis de 100 Números", "Cálculo de 2 Números" y "Cálculo de Serie de Taylor". Cada botón está vinculado a una ruta específica que redirige a la pantalla correspondiente.

```
1 import 'package:flutter/material.dart';
2
3 class PaginaInicial extends StatefulWidget {
4   const PaginaInicial({super.key});
5   @override
6   State<PaginaInicial> createState() => _PaginaInicialState();
7 }
8
9 class _PaginaInicialState extends State<PaginaInicial> {
10   @override
11   Widget build(BuildContext context) {
12     return Scaffold(
13       appBar: AppBar(
14         title: const Text('AlexGames Studios'),
15         backgroundColor: Colors.lightBlueAccent,
16       ),
17       body: Center(
18         child: Column(
19           children: <Widget>[
20             const Padding(
21               padding: EdgeInsets.only(top: 60, bottom: 20),
22               child: Text(
23                 'Bienvenido a AlexGames Studios!',
24                 style: TextStyle(fontSize: 24, fontWeight: FontWeight.bold),
25                 textAlign: TextAlign.right,
26               ),
27             ),
28             const SizedBox(height: 60),
29             _buildBotonRuta(context, 'Cálculo de Sueldo', '/calculo_sueldo'),
30             const SizedBox(height: 20),
31             _buildBotonRuta(context, 'Análisis de 100 Números', '/analisis_100'),
32             const SizedBox(height: 20),
33             _buildBotonRuta(context, 'Cálculo de 2 Números', '/calculo_2_numeros'),
34             const SizedBox(height: 20),
35             _buildBotonRuta(context, 'Cálculo de Serie de Taylor', '/serie_taylor'),
36           ],
37         ),
38       ),
39     );
40   };
41 }
```

```

42 Widget _buildBotonRuta(BuildContext context, String texto, String ruta) {
43   return SizedBox(
44     width: 400,
45     child: ElevatedButton(
46       style: ElevatedButton.styleFrom(
47         minimumSize: const Size.fromHeight(50),
48       ),
49       onPressed: () {
50         Navigator.pushNamed(context, ruta);
51       },
52       child: Text(texto),
53     ),
54   );
55 }
56 }
57

```

8. Este Código (main.dart) define la estructura de una aplicación Flutter llamada "AlexGames Studios". La aplicación utiliza MaterialApp para configurar el tema y las rutas de navegación. Al iniciar la aplicación, se muestra la pantalla inicial (PaginaInicial) como la ruta predeterminada.

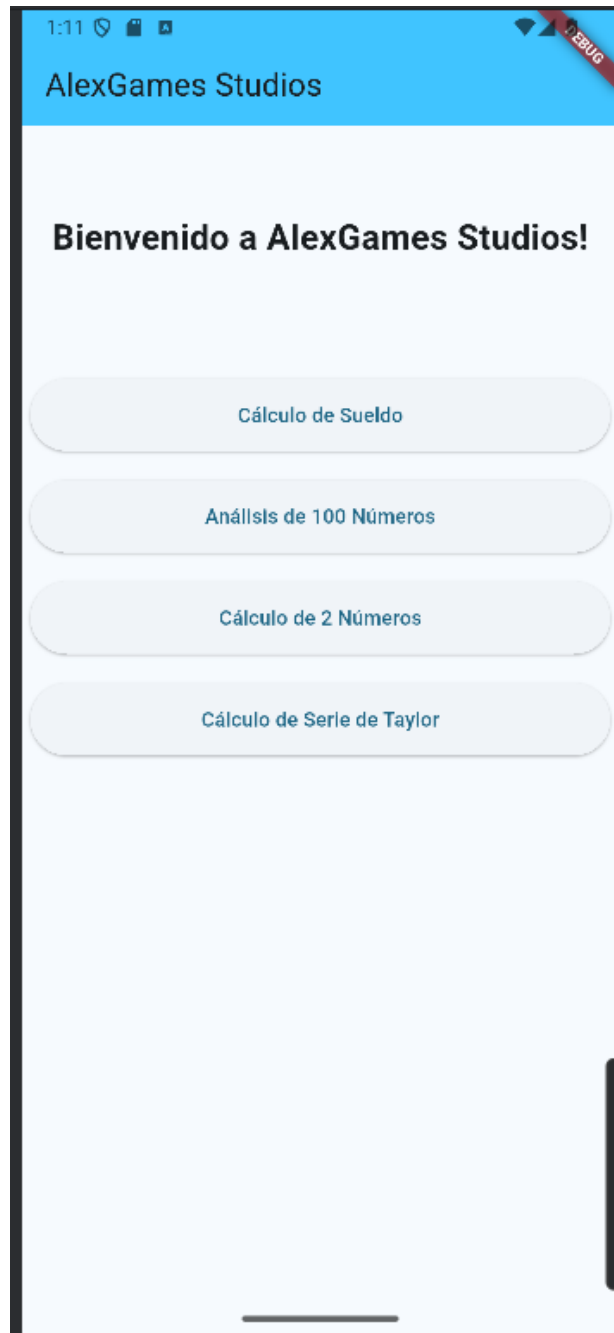
```

1 import 'package:flutter/material.dart';
2 import 'package:tarea_1_1/pantallas/analisis_100_numeros.dart';
3 import 'package:tarea_1_1/pantallas/calculo_2_numeros.dart';
4 import 'package:tarea_1_1/pantallas/calculo_serie_taylor.dart';
5 import 'package:tarea_1_1/pantallas/calculo_sueldo.dart';
6 import 'package:tarea_1_1/pantallas/pagina_inicial.dart';
7
8 void main() {
9   runApp(const MyApp());
10 }
11
12 class MyApp extends StatelessWidget {
13   const MyApp({super.key});
14
15   @override
16   Widget build(BuildContext context) {
17     return MaterialApp(
18       title: 'AlexGames Studios',
19       theme: ThemeData(
20         colorScheme: ColorScheme.fromSeed(seedColor: Colors.lightBlueAccent),
21       ),
22       initialRoute: '/',
23       routes: {
24         '/': (context) => const PaginaInicial(),
25         '/calculo_sueldo': (context) => CalculoSuelto(),
26         '/analisis_100': (context) => Analisis100Numeros(),
27         '/calculo_2_numeros': (context) => Calculo2Numeros(),
28         '/serie_taylor': (context) => CalculoSerieTaylor(),

```

V. Resultados

1. Página inicial



2. Calcular sueldo



The screenshot shows a mobile application interface for calculating salary. At the top, there is a blue header bar with a back arrow, the title "Cálculo de Sueldo", and a red "DEBUG" banner. Below the header, the text "Ingrese los datos del empleado:" is displayed. There are two input fields: "Edad" with the value "25" and "Años en la empresa" with the value "5". A "Calcular sueldo" button is positioned below the input fields. The result, "El sueldo semanal es: \$36525", is shown in bold text. At the bottom, there are three buttons: "← Atrás", "🏠 Inicio", and "→ Siguiente". The status bar at the very top shows the time "1:11" and various system icons.

1:11

← Cálculo de Sueldo

DEBUG

Ingrese los datos del empleado:

Edad 25

Años en la empresa 5

Calcular sueldo

El sueldo semanal es: \$36525

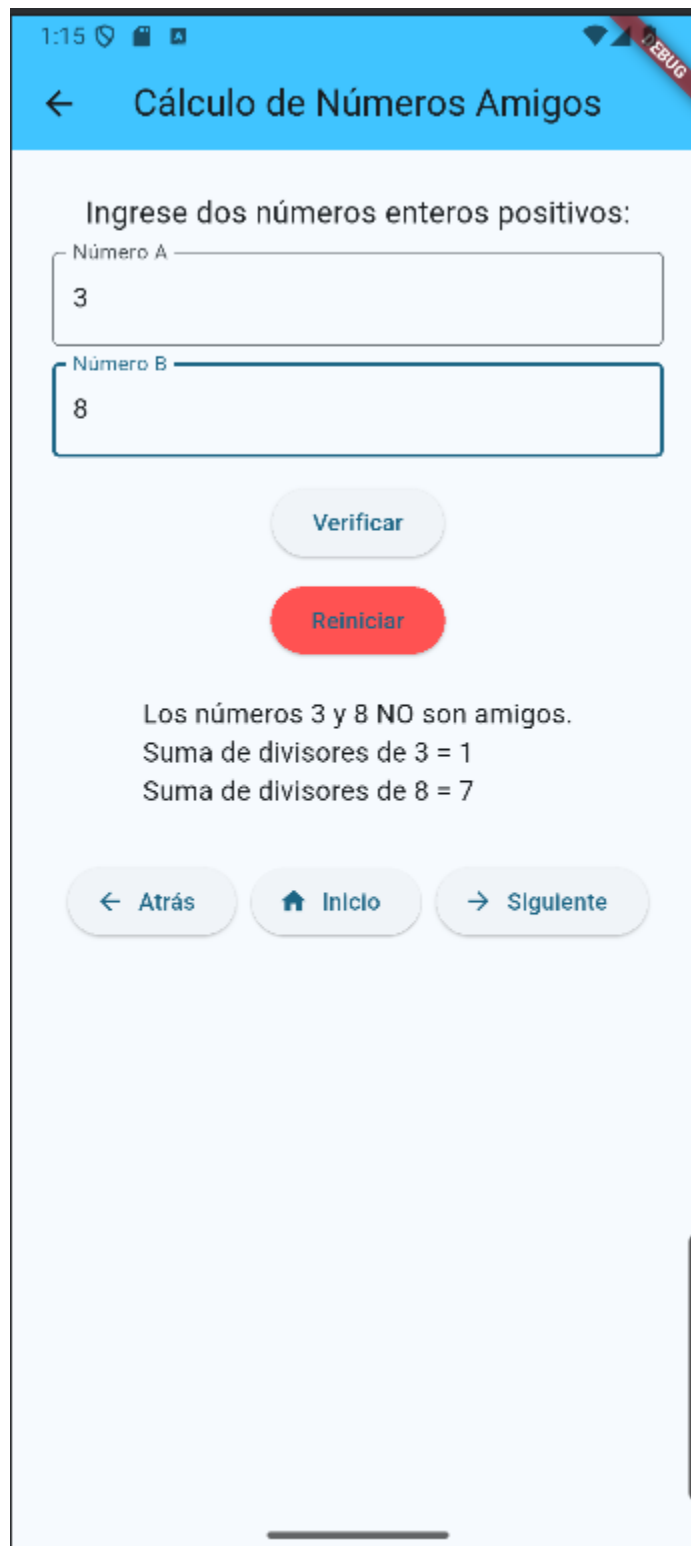
← Atrás 🏠 Inicio → Siguiente

3. Análisis de 100 números



The screenshot shows a mobile application interface with a blue header bar. The title 'Análisis de 100 Números' is displayed in white, preceded by a back arrow icon. In the top right corner of the header, there is a red 'DEBUG' banner. Below the header, the text 'Ingrese número 4 de 100:' is centered. Underneath this text is a text input field with a light blue border and a placeholder label 'Número natural' at the top left. Below the input field are three buttons: 'Agregar número(s)' (light blue), 'Calcular resultados' (light blue), and 'Reiniciar todo' (red). Below these buttons, a message reads 'Debes ingresar exactamente 100 números antes de calcular.' At the bottom of the screen, there are three navigation buttons: '← Atrás' (light blue), 'Inicio' (light blue with a house icon), and '→ Siguiete' (light blue). The status bar at the very top shows the time '1:13' and various system icons.

4. Cálculo números amigos



1:15

← Cálculo de Números Amigos

INGRESE DOS NÚMEROS ENTEROS POSITIVOS:

Número A
3

Número B
8

Verificar

Reiniciar

Los números 3 y 8 NO son amigos.
Suma de divisores de 3 = 1
Suma de divisores de 8 = 7

← Atrás Inicio → Sigulente

VII. Conclusiones

El desarrollo de una aplicación educativa utilizando Flutter ha demostrado ser una excelente forma de integrar la programación con el aprendizaje interactivo. Mediante la implementación de diversas funcionalidades matemáticas, como el cálculo de sueldos, análisis de números y series de Taylor, se ha facilitado la comprensión de conceptos complejos a través de una interfaz visual y fácil de usar.

Las interfaces diseñadas para que los usuarios ingresen datos y reciban resultados en tiempo real mejoran significativamente la experiencia de aprendizaje. La posibilidad de interactuar con la aplicación y obtener resultados instantáneos promueve un enfoque más práctico y comprensible de los conceptos matemáticos.

La creación de una aplicación con múltiples funcionalidades matemáticas en Flutter ha proporcionado una comprensión sólida de cómo estructurar aplicaciones móviles, manejar rutas de navegación, gestionar entradas de usuario y realizar cálculos. Estas habilidades son fundamentales no solo para el desarrollo de aplicaciones educativas, sino también para la construcción de soluciones tecnológicas en diversas áreas.

Referencias

[1] Brown, J. (2009). *Amicable numbers*. Occidental College. Recuperado de <https://jim-brown.oxy.edu/Teaching/Math573/Math573amicablenumbers.pdf>

[2] Chatterjee, S. (2021). *Exploring the role of Taylor series in machine learning: From function approximation to model optimization*. Medium.
<https://medium.com/aimonks/exploring-the-role-of-taylor-series-in-machine-learning-from-function-approximation-to-model-30aaf612113f>

[3] Kumar, A., & Singh, R. (2023). *Analysis of cross platform application development over multiple devices using Flutter & Dart*. ResearchGate.
<https://www.researchgate.net/publication/371157369>

[4] Tuna, A., & Tuna, D. (2020). The effect of algorithm education on students' computer programming self-efficacy perceptions and computational thinking skills. *Education and Information Technologies*, 25(3), 2673–2689.
<https://www.researchgate.net/publication/339673683>