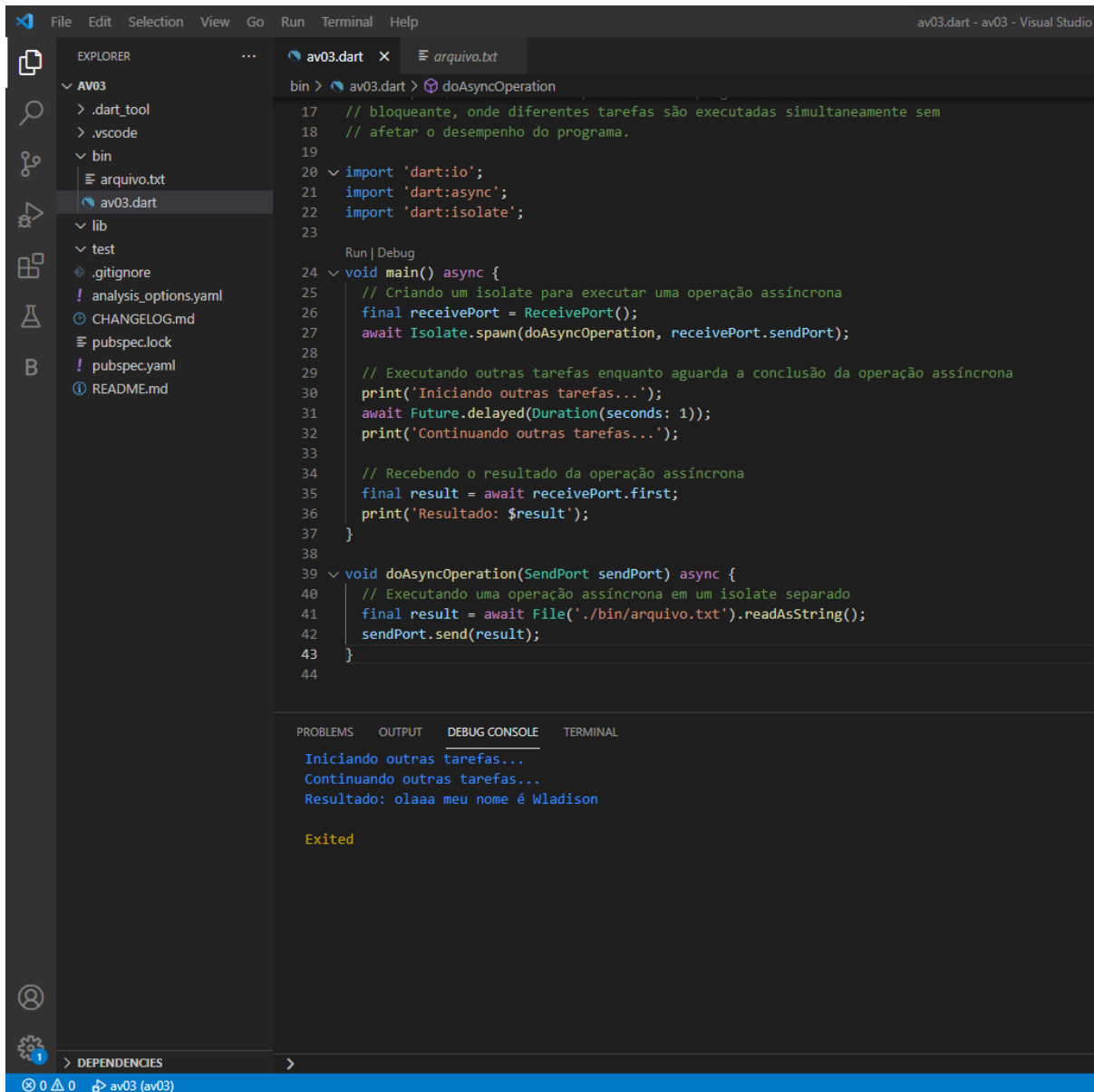


## Atividade 03 - Programação de Dispositivos Móveis



The screenshot shows the Visual Studio Code interface with a Dart project named 'av03'. The Explorer sidebar on the left shows the project structure, including a 'bin' directory with 'av03.dart' and 'arquivo.txt'. The main editor displays the code in 'av03.dart', which uses Dart's isolate mechanism to run asynchronous tasks without blocking the main thread. The code includes imports for 'dart:io', 'dart:async', and 'dart:isolate'. The 'main' function creates a 'ReceivePort' and spawns a 'doAsyncOperation' isolate. This isolate reads the content of 'arquivo.txt' and sends it back to the main thread via the 'SendPort'. The output in the Debug Console shows the sequence of events: starting other tasks, continuing other tasks, and finally receiving the result 'olaaa meu nome é Wladison'.

```
17 // bloqueante, onde diferentes tarefas são executadas simultaneamente sem
18 // afetar o desempenho do programa.
19
20 import 'dart:io';
21 import 'dart:async';
22 import 'dart:isolate';
23
24 void main() async {
25   // Criando um isolate para executar uma operação assíncrona
26   final receivePort = ReceivePort();
27   await Isolate.spawn(doAsyncOperation, receivePort.sendPort);
28
29   // Executando outras tarefas enquanto aguarda a conclusão da operação assíncrona
30   print('Iniciando outras tarefas...');
31   await Future.delayed(Duration(seconds: 1));
32   print('Continuando outras tarefas...');
33
34   // Recebendo o resultado da operação assíncrona
35   final result = await receivePort.first;
36   print('Resultado: $result');
37 }
38
39 void doAsyncOperation(SendPort sendPort) async {
40   // Executando uma operação assíncrona em um isolate separado
41   final result = await File('./bin/arquivo.txt').readAsString();
42   sendPort.send(result);
43 }
44
```

DEBUG CONSOLE

```
Iniciando outras tarefas...
Continuando outras tarefas...
Resultado: olaaa meu nome é Wladison

Exited
```