```dart
11    // para o isolate principal através do ReceivePort criado anteriormente.
12
13    // Esse exemplo demonstra como os isolates podem ser usados em conjunto
14    // com operações assíncronas para criar um programa concorrente e não
15    // bloqueante, onde diferentes tarefas são executadas simultaneamente sem
16    // afetar o desempenho do programa.
17
18    import 'dart:io';
19    import 'dart:async';
20    import 'dart:isolate';
21
      Run | Debug
22    void main() async {
23      // Criando um isolate para executar uma operação assíncrona
24      final receivePort = ReceivePort();
25      await Isolate.spawn(doAsyncOperation, receivePort.sendPort);
26
27      // Executando outras tarefas enquanto aguarda a conclusão da operação assíncrona
28      print('Iniciando outras tarefas...');
29      await Future.delayed(Duration(seconds: 1));
30      print('Continuando outras tarefas...');
31
32      // Recebendo o resultado da operação assíncrona
33      final result = await receivePort.first;
34      print('Resultado: $result');
35    }
36
37    void doAsyncOperation(SendPort sendPort) async {
38      // Executando uma operação assíncrona em um isolate separado
39      final result = await File('./bin/arquivo.txt').readAsString();
40      sendPort.send(result);
41    }
```

PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS

```
Iniciando outras tarefas...
Continuando outras tarefas...
Resultado: Oi, eu sou o Pedro.

Exited.
```