

atividades assíncronas em dart

Nível Fácil

- 1 - Crie uma função assíncrona chamada `buscaDados` que simule uma busca de dados com um atraso de 1 segundo usando `Future.delayed`.
- 2 - Escreva uma função chamada `multiplicar` que receba dois números como parâmetros e retorne o produto deles de forma assíncrona.
- 3 - Implemente uma função assíncrona que faz uma chamada a uma API fictícia usando `http.get` e retorne o resultado.
- 4 - Crie uma função que utilize `Future.wait` para esperar duas operações assíncronas serem concluídas e depois imprima os resultados.
- 5 - Escreva um código que chama uma função assíncrona e utiliza `await` para esperar pela conclusão antes de imprimir "Fim".
- 6 - Faça uma função que retorna uma lista de números de 1 a 5 de forma assíncrona usando `Future.delayed`.
- 7 - Implemente uma função assíncrona que simula a leitura de um arquivo e retorna o conteúdo após um atraso de 2 segundos.
- 8 - Crie uma função que receba um nome e retorne uma saudação após um atraso de um segundo.

9 - Desenvolva um programa que chame uma função assíncrona que conta de 1 a 3 com um atraso de 1 segundo entre os números.

10 - Escreva uma função que faz uma operação assíncrona de soma de dois números e retorna o resultado.

Nível Médio:

1 - Crie uma função assíncrona chamada `processarDados` que simule o processamento de dados e retorne uma mensagem após 3 segundos.

2 - Implemente uma função que busca dados de uma lista assíncrona e retorna apenas os números pares.

3 - Escreva uma função que utiliza `try-catch` para lidar com erros em uma operação assíncrona.

4 - Desenvolva um programa que chama duas funções assíncronas e utiliza `Future.wait` para esperar ambas terminarem antes de prosseguir.

5 - Crie uma função assíncrona que simula a busca de um usuário em um banco de dados e retorna informações fictícias.

6 - Faça uma função que retorna uma lista de strings de forma assíncrona e as imprime uma a uma após um atraso de 1 segundo.

7 - Escreva um programa que lê e exibe um arquivo de texto de forma assíncrona utilizando a biblioteca `dart:io`.

8 - Implemente uma função que simula uma operação de login com um atraso e retorna um valor booleano indicando sucesso ou falha.

9 - Crie um código que chama uma função assíncrona que espera um número do usuário e imprime se ele é par ou ímpar.

10 - Desenvolva uma função que faz uma chamada assíncrona a uma API e retorna um objeto JSON após um atraso.

Nível Difícil:

1 - Crie uma aplicação que utiliza `async/await` para gerenciar um fluxo de trabalho que envolve múltiplas operações assíncronas.

2 - Implemente uma função que busca dados de múltiplas APIs e combina os resultados em um único objeto.

3 - Escreva um código que utiliza `async` para realizar operações de leitura e escrita em um banco de dados de forma assíncrona.

4 - Desenvolva uma aplicação que consome uma API de clima e exibe a temperatura atual de uma cidade de forma assíncrona.

5 - Crie um sistema que simula um carrinho de compras, onde cada adição de item é feita de forma assíncrona.

6 - Implemente um código que utiliza `streams` assíncronos para processar uma lista de dados em tempo real.

7 - Desenvolva uma função que faz uma chamada assíncrona a um serviço de pagamento e

retorna o status da transação.

8 - Crie uma aplicação que utiliza `async/await` para gerenciar uma fila de tarefas a serem executadas em sequência.

9 - Escreva um código que integra múltiplas operações assíncronas com a utilização de `Completer`.

10 - Implemente uma função que faz a autenticação de um usuário de forma assíncrona, tratando erros de forma adequada.