

## **1. Programação Síncrona**

Uma estrutura de código onde que os mesmos são executados sequencialmente na ordem que foram escritos pelo desenvolvedor.

As linhas de comando só serão executadas com a finalização da execução que há antecede. Caso um método/variável/classe dependa do resultado de outra, esta ficará parada até que o resultado seja retornando ao sistema, até que isso aconteça o programa interior para de funcionar da perspectiva do usuário, ocasionando uma má utilização do processamento da máquina.

## **2. Programação Assíncrona**

Esta vem para resolver o problema da programação síncrona, ou seja, cada linha de comando será executada sem que ocorra o término da antecessora, sendo assim, a etapa 2 será executada mesmo que a etapa 1 não seja concluída.

Programação Assíncrona serve para agilizar os processos de execução de um programa, pois cada comando será executado sem o término do comando anterior.

### **2.1 Future**

É uma classe que faz com que um método/variável retorne um resultado no ‘futuro’, sendo assim, uma função que levará um tempo determinado para sua finalização.

### **2.2 Await**

É utilizada para determinar que o aplicativo deve esperar uma resposta de uma função antes de continuar sua execução, pois existem casos que uma função só consegue prosseguir havendo o resultado/retorno de uma outra.

### **2.3 Async**

Determina/nomeia quando um método será assíncrono, ou seja, não terá um resultado imediatamente, onde a aplicação pode continuar a execução de outras tarefas enquanto o processamento não é finalizado.