

 02

A importância de fechar um recurso

Vimos que é importante fechar o arquivo, pois liberamos o recurso que está sendo utilizado pela aplicação.

Podemos imaginar a liberação de recursos, por exemplo, quando pessoas deixam um restaurante. A mesa (o recurso) está livre para ser utilizada por outras pessoas (os processos). Caso nenhuma mesa esteja disponível, as pessoas que querem comer no restaurante terão que ficar esperando – no mundo da computação, causando lentidão na execução do processo – ou vão embora – causando um erro na execução do programa.

Portanto, a boa prática de programação diz para sempre liberarmos um recurso logo após seu uso.

Fechamos o arquivo para liberá-lo no sistema operacional, porém, existem muitos outros recursos que também podem, e precisam, ser fechados como, por exemplo, a conexão com banco de dados e *sockets* de rede.

Bancos de dados, por exemplo, podem ter um número restrito de conexões que podem ser abertas, algo conhecido como *pool* de conexões. Quando todas essas conexões estão em uso, não é possível fazer uma nova conexão para manipular os dados. Logo, percebemos quão importante é liberar um recurso.