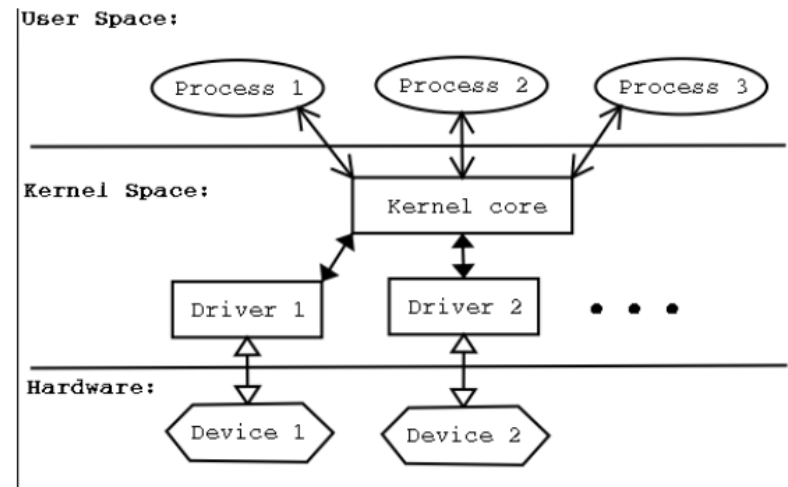


# Componentes Típicos de um S.O.

- **Núcleo (kernel) e drivers:**
  - Executam as funções mais básicas para alocação e gerenciamento seguro dos recursos
  - Acessam diretamente o hardware e todas as instruções da máquina (modo kernel)
  - Drivers: fazem o controle específico de uma (classe de) dispositivos de Entrada e Saída.



# Operação do sistema de computação

- Conhecer a **estrutura dos sistemas de computação** é item primordial antes que possamos explorar os detalhes de seu funcionamento. Os **itens básicos** contemplam:
  - **Inicialização do sistema.**
  - **Entrada e saída (*input/output* – I/O).**
  - **Armazenamento.**
- Um moderno sistema de computação de uso geral compõe-se basicamente de:
  - um processador (CPU);
  - controladores de dispositivos;
  - um *bus* comum.

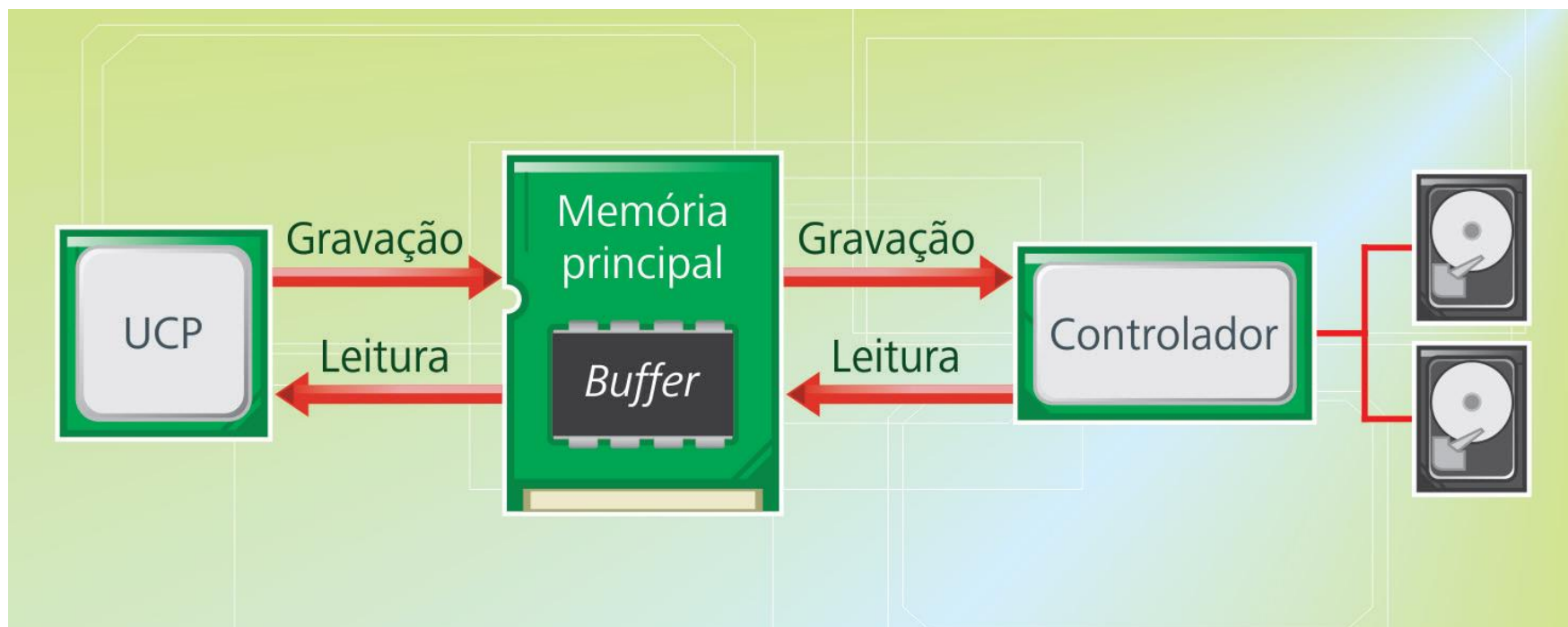
# Interrupções e exceções

- Enquanto um programa é executado, podem ocorrer alguns **eventos inesperados, ocasionando um desvio forçado do seu fluxo de execução**. A estes eventos, denomina-se **de interrupção ou exceção**. Tanto a interrupção como a exceção podem ocorrer em consequência da sinalização de algum dispositivo de *hardware* externo à CPU ou da execução de instruções do próprio programa.
- **Uma interrupção é sempre causada por um evento externo ao programa que está em execução, ou seja, independe da instrução que está sendo executada.**
- Um exemplo de interrupção é quando **um dispositivo de *hardware* avisa ao processador que alguma operação de E/S está completa**. Neste caso, a CPU deve interromper suas atividades atuais no programa em execução para tratar o término da operação de E/S. A Figura abaixo, demonstra o mecanismo de interrupção e exceção de forma ilustrativa.

- **Uma exceção** é bastante semelhante a uma interrupção, sendo o que diferencia uma da outra é o motivo pelo qual o evento é gerado. Uma exceção pode ocorrer **através de uma instrução do próprio *software***, como por exemplo, a divisão de um número por zero, entre outros. A exceção diferentemente da interrupção é gerada por um evento síncrono (resultado direto da execução do programa corrente).
- **Na grande maioria das vezes, uma exceção provoca um erro fatal no sistema, causando o término anormal do programa. Recomenda-se que a exceção deve ser tratada dentro do próprio programa, com instruções escritas pelo programador.**
- Da mesma forma que na interrupção, **sempre que uma exceção é gerada o programa em execução é interrompido e o controle é desviado para uma rotina de tratamento de exceção.**

# ***Buffering***

- A técnica denominada de *buffering* consiste em utilizar uma área de memória principal, chamada *buffer*, criada e mantida pelo sistema operacional. Possui a finalidade de auxiliar a transferência de dados entre dispositivos de E/S e a memória. O *buffer* permite minimizar a disparidade de velocidade entre o processador e os dispositivos de E/S e, tem como objetivo principal, manter tanto os dispositivos de E/S como o processador, ocupados a maior parte do tempo.
- O registro é a unidade de transferência do mecanismo de *buffering*. O *buffer* deve comportar o armazenamento de diversos registros, de forma que o processador tenha à sua disposição dados suficientes para processar sem ter que interromper o programa a cada leitura/gravação no dispositivo de E/S.
- Na Figura abaixo, é possível visualizar como as operações de entrada e saída utilizam o *buffer*.

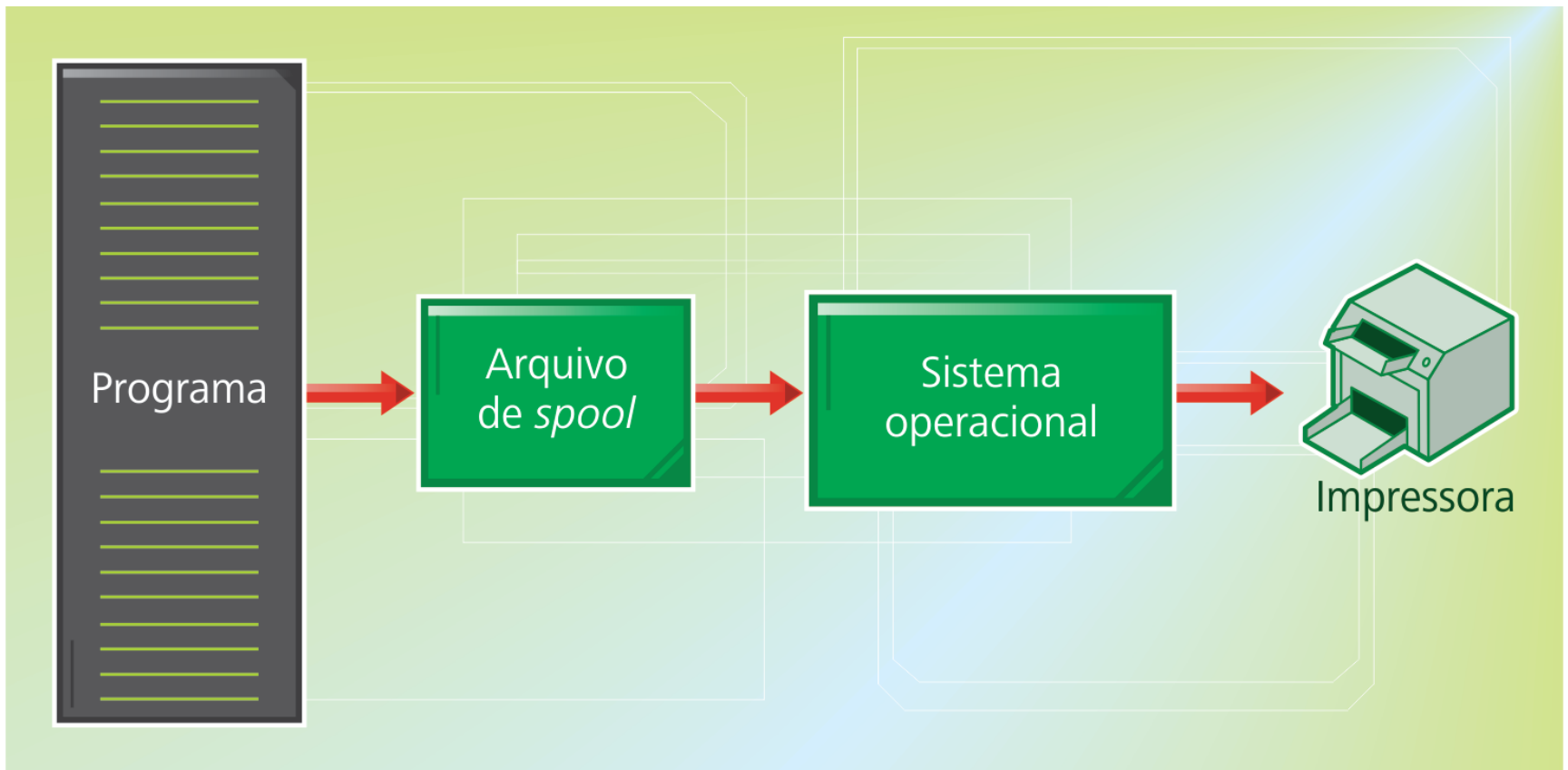


**Operação de entrada e saída utilizando *buffer***

Fonte: CTISM, adaptado de Maia, 2007

# *Spooling*

- A técnica de *spooling* foi criada inicialmente para auxiliar a submissão de processos ao sistema, sendo os processos gravados em fita para posterior leitura e execução. Com o aparecimento dos terminais para acesso ao sistema, esta técnica teve sua função adaptada para armazenar o resultado da impressão dos programas em execução.
- Isto é possível através da criação e manutenção, pelo **sistema operacional de uma grande área em disco, com a finalidade de simular uma impressora.**
- Assim, os **usuários e seus programas imprimem, na verdade, para este arquivo em disco, liberando a associação dos dispositivos de impressão diretamente aos programas que estão executando.**



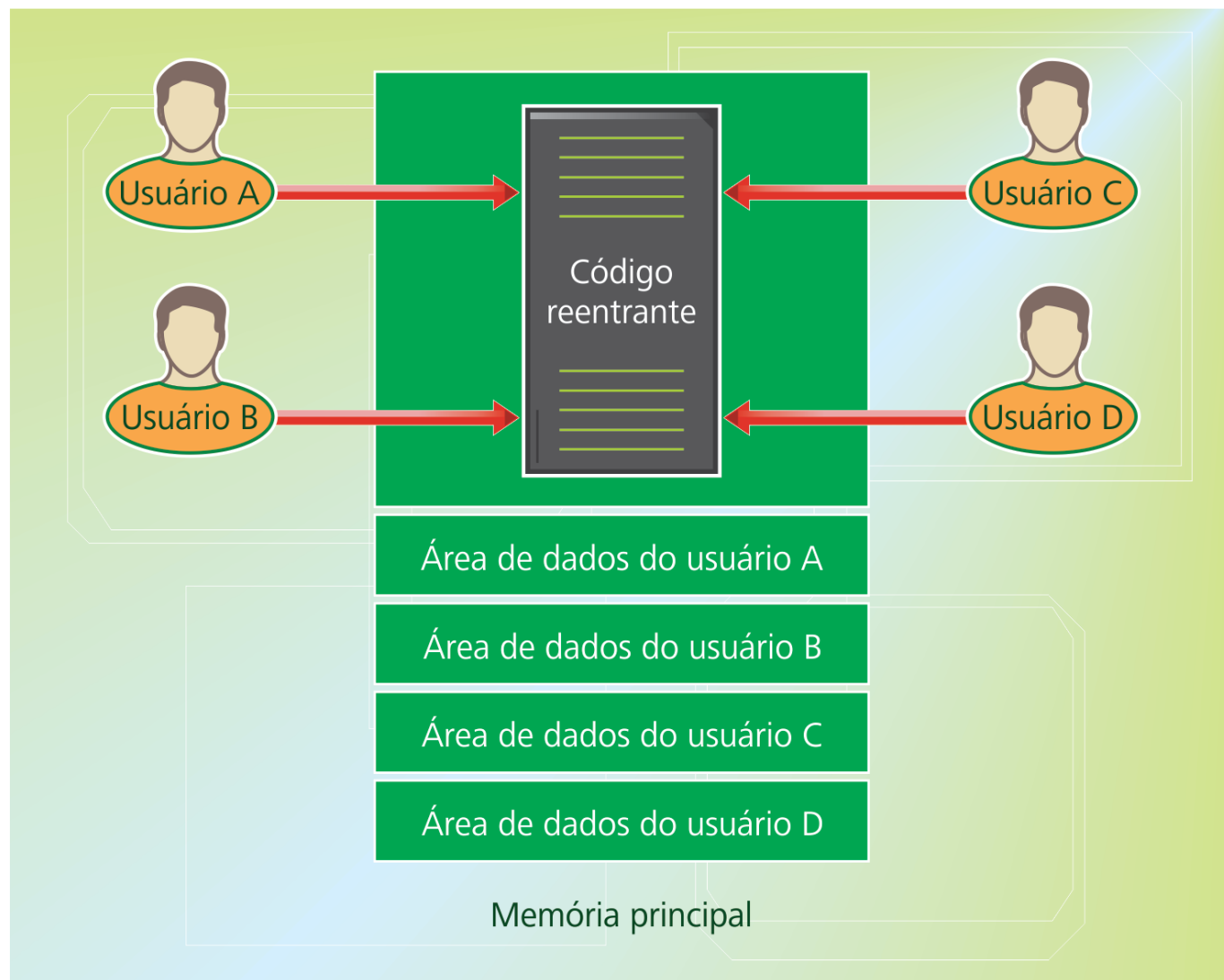
## **Técnica de *spooling***

Fonte: CTISM, adaptado de Maia, 2007



# Reentrância

- Em sistemas multiprogramáveis, é normal que vários usuários utilizem os mesmos aplicativos simultaneamente, como editores de texto, compiladores, entre outros utilitários.
- Nestes casos, se cada usuário que fosse utilizar um destes aplicativos trouxesse o código executável para a memória haveria então diversas cópias de um mesmo programa ocupando espaço na memória, o que causaria um grande desperdício de espaço.
- **A reentrância é a capacidade de um código executável (código reentrante) ser compartilhado por vários usuários, exigindo apenas uma cópia do programa em memória.** Esta técnica permite que cada usuário esteja executando um trecho diferente do código reentrante, manipulando dados próprios, exclusivos de cada usuário.
- Assim, a reentrância tem como objetivo geral promover o uso mais eficiente da memória e um desempenho maior do sistema. A Figura abaixo, ilustra o cenário da reentrância.

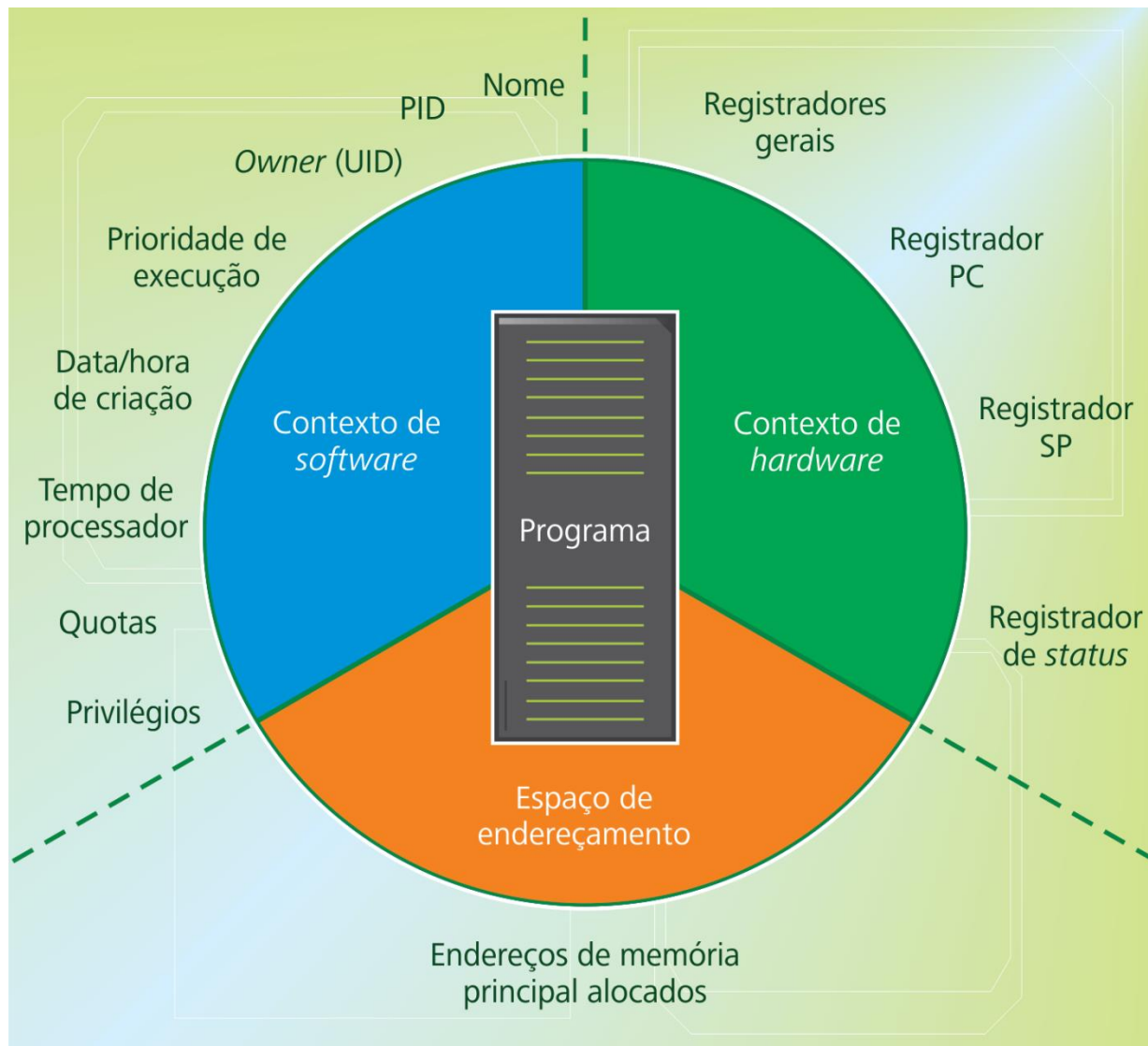


## Reentrância

Fonte: CTISM, adaptado de Maia, 2007

# Gerenciamento de processos

- Podemos definir um **processo** em sistemas operacionais como um programa em execução.
- Enquanto que um **processo** é uma entidade **ativa**, ou seja, algo que **entrou em execução** e **que possui um conjunto de recursos associados a ele**. Um sistema operacional é constituído, portanto, por uma coleção de processos, que se subdividem em processos do sistema operacional e processos de usuário.



### Características da estrutura de um processo

Fonte: CTISM, adaptado de Maia, 2007

# Estados do processo



- Um processo muda de estado durante a sua execução.
- Cada processo pode estar em cada um dos seguintes estados:
  - **Novo** – o processo está sendo criado.
  - **Execução** – as instruções estão sendo executadas.
  - **Espera** – o processo está esperando pela ocorrência de algum evento (como um término de I/O).
  - **Pronto** – o processo está esperando para ser designado a um processador.
  - **Terminado** – o processo terminou a sua execução.



# **Gerenciador de Tarefas do Windows 10 - Dicas e Truques**

[Gerenciador de Tarefas do Windows 10 - Dicas e Truques – YouTube](#)

[Gerenciador de Tarefas do Windows 10 - Dicas e Truques.mp4](#)