**Solução para o problema**

Temos que criar uma *nova condicional SE*.

A nova condicional deve testar **antes de tudo**, se os números são iguais.

Caso isso seja Verdadeiro, mostra a mensagem de "os números são iguais".

Caso contrário (senão), vai efetuar os testes da outra condicional SE.

**Dicas**

**Sobre as variáveis:**

- Declare apenas 2 variáveis para esse algoritmo. Elas devem ser do tipo *real* e podem ser chamadas de **numero** e **menor**.

**Sobre o programa:**

- Ao receber o primeiro número, faça com que o seu valor seja atribuído para a variável **menor**  
- Depois, **a cada número recebido** compare se o novo número é menor ( < ) que o número da variável menor  
- Caso seja Verdadeiro, então o novo número passa a ser o menor número (atribuir para a variável menor)

**ESTRUTURA DE DECISÃO: CASO**

**Explicação**

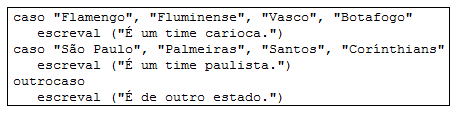
Iremos aprender agora sobre a estrutura de decisão do tipo **CASO**.

Nas etapas anteriores, utilizamos a condição SE para criar os desvios no código. A condição SE sempre atende a uma condição, e dependendo do resultado do teste lógico, o programa segue pelo caminho desejado.

Utilizando a estrutura CASO, podemos determinar vários caminhos, que será escolhido através de uma expressão. Por exemplo: podemos determinar que o programa escolha qual caminho seguir, dependendo de um número informado, o usuário poderia ter 5 ou "n" caminhos diferentes.

A estrutura CASO também pode ser chamada de "escolha".

Veja um exemplo:

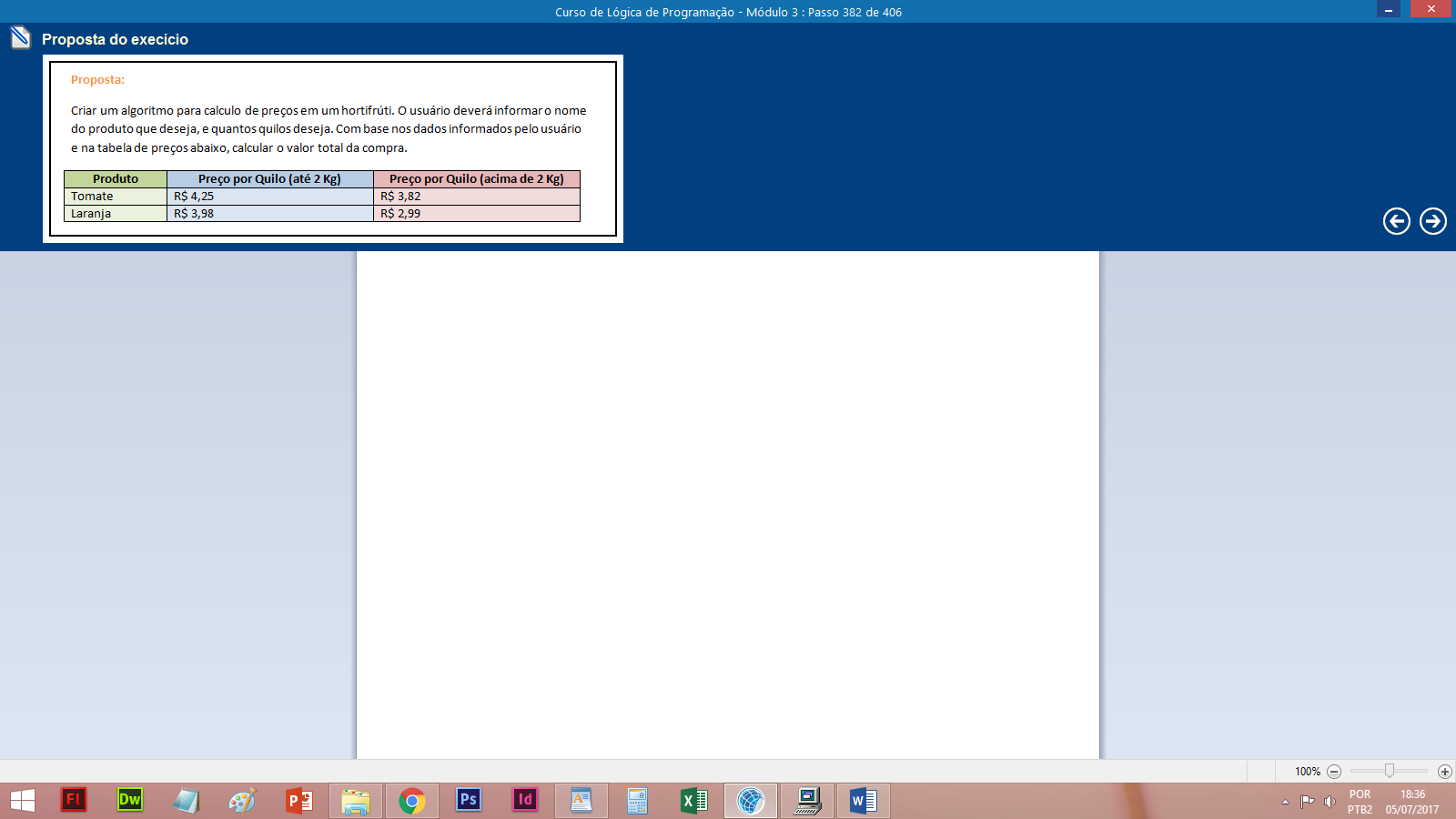
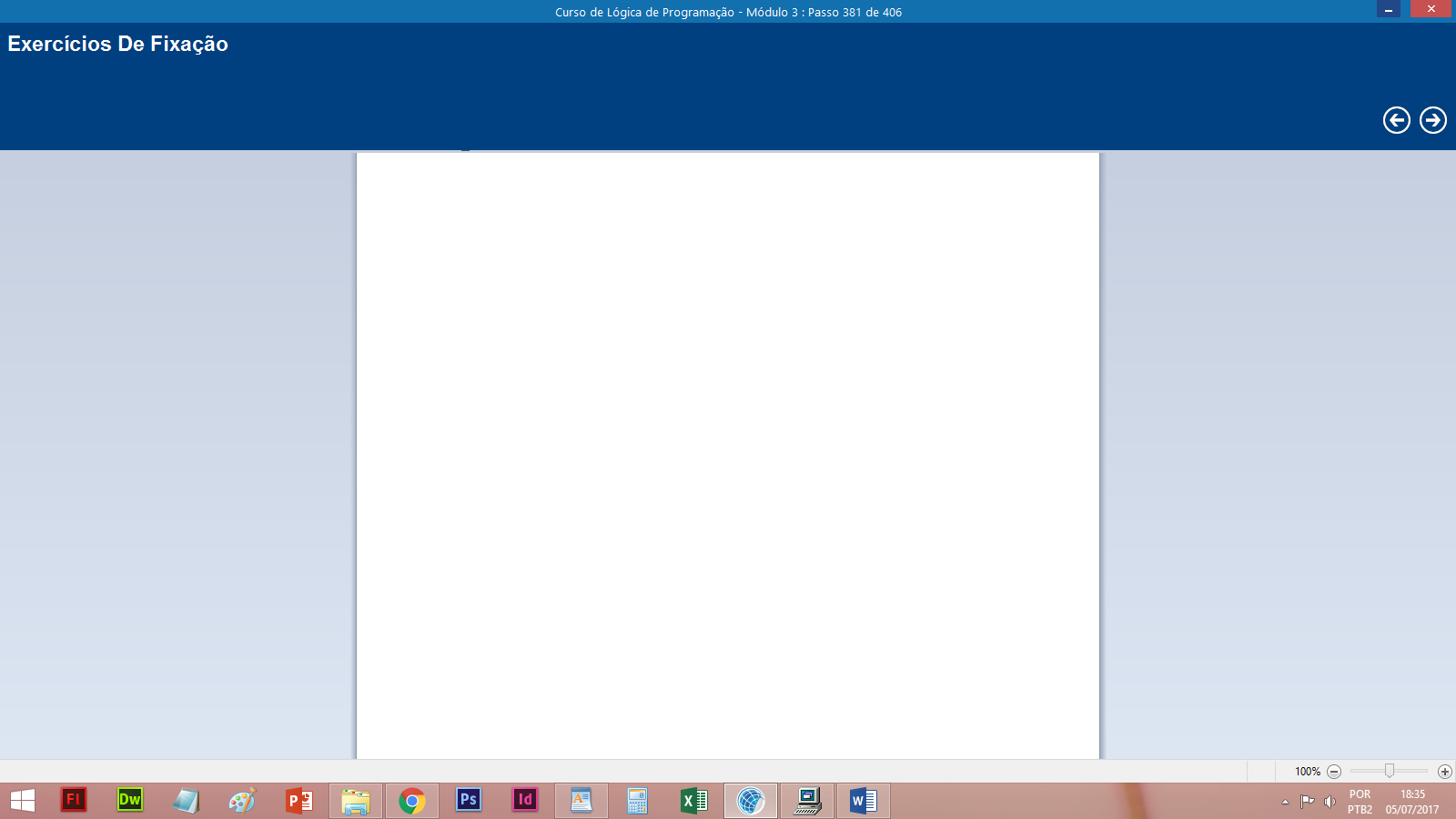


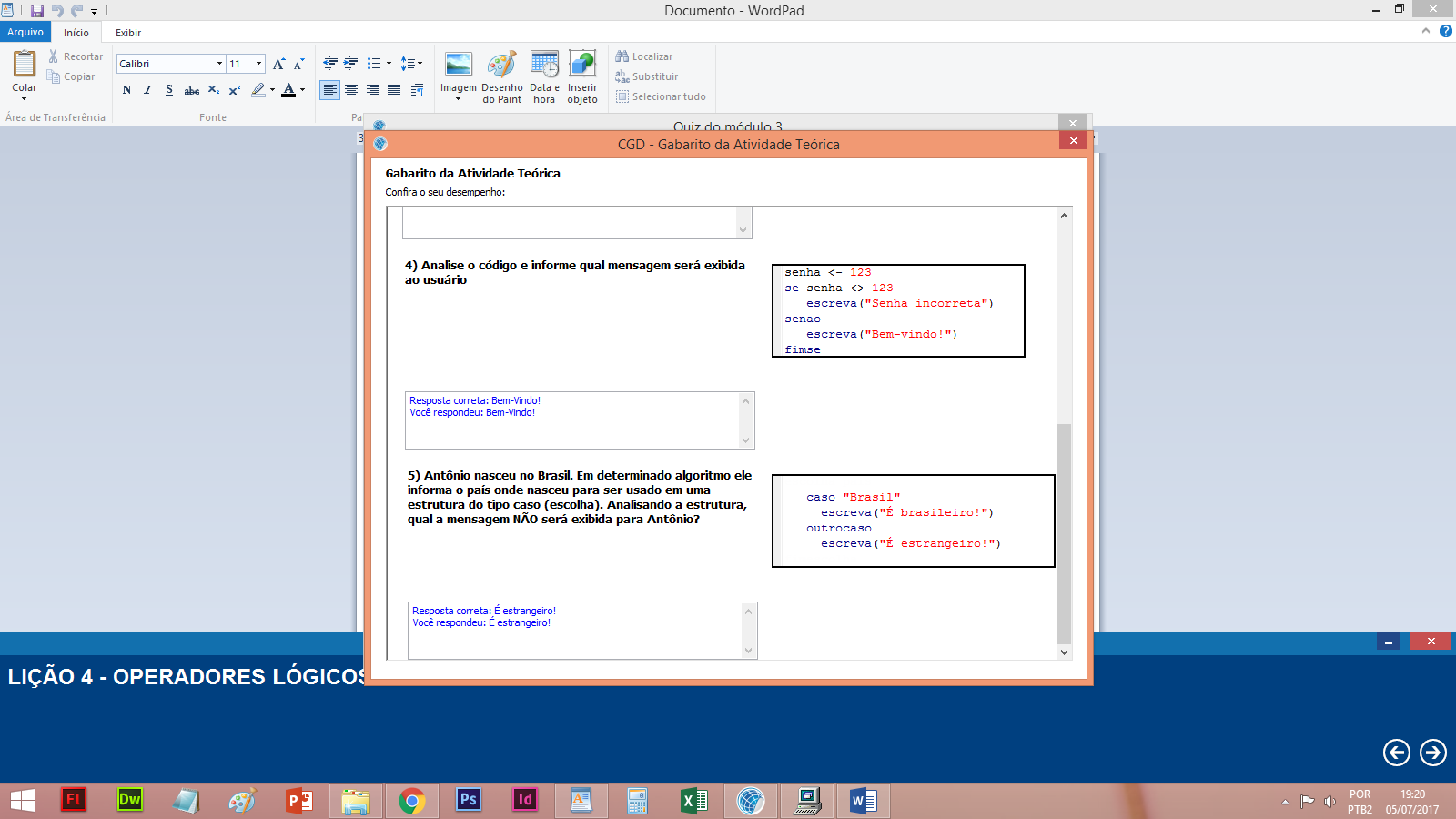
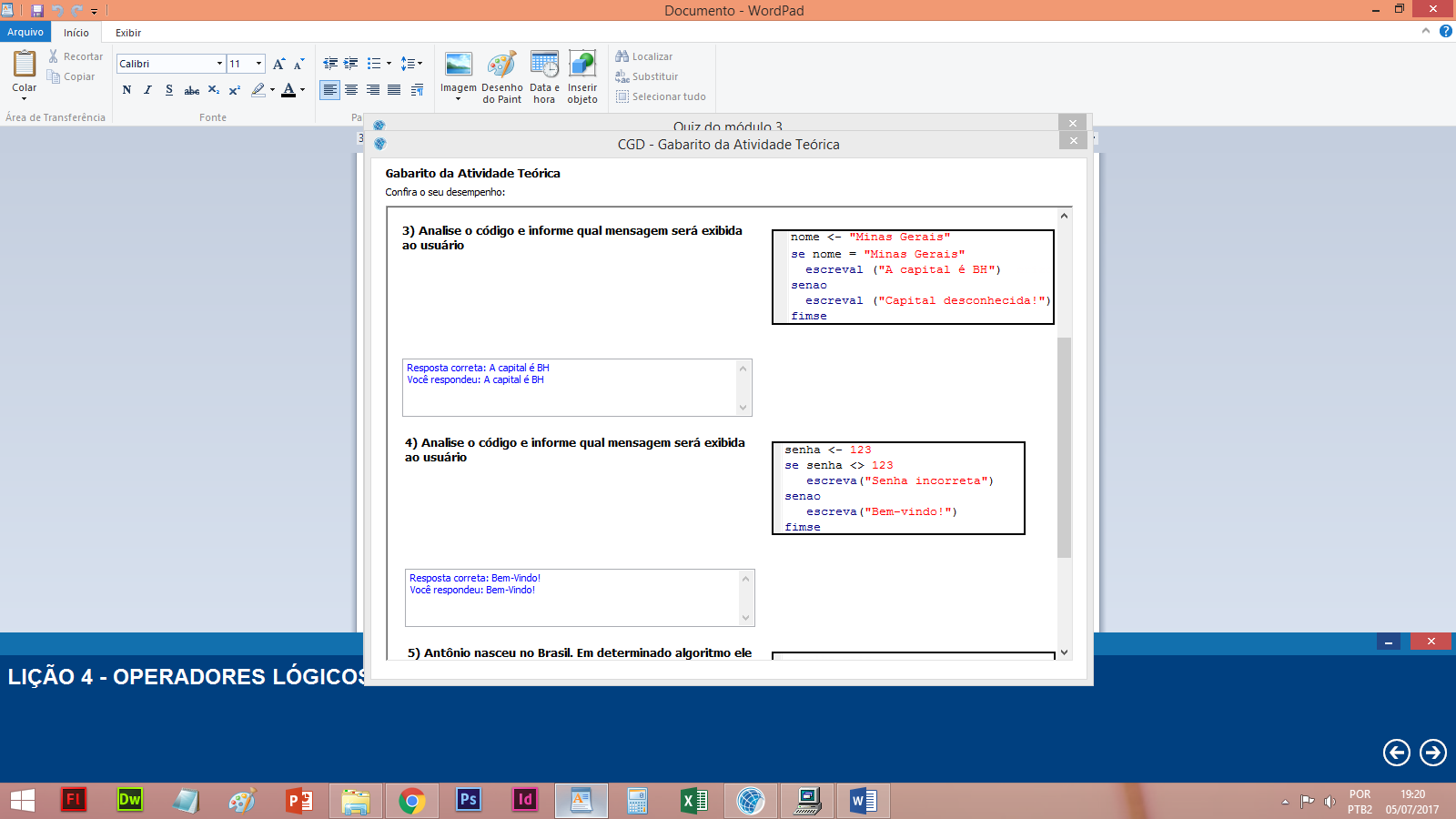
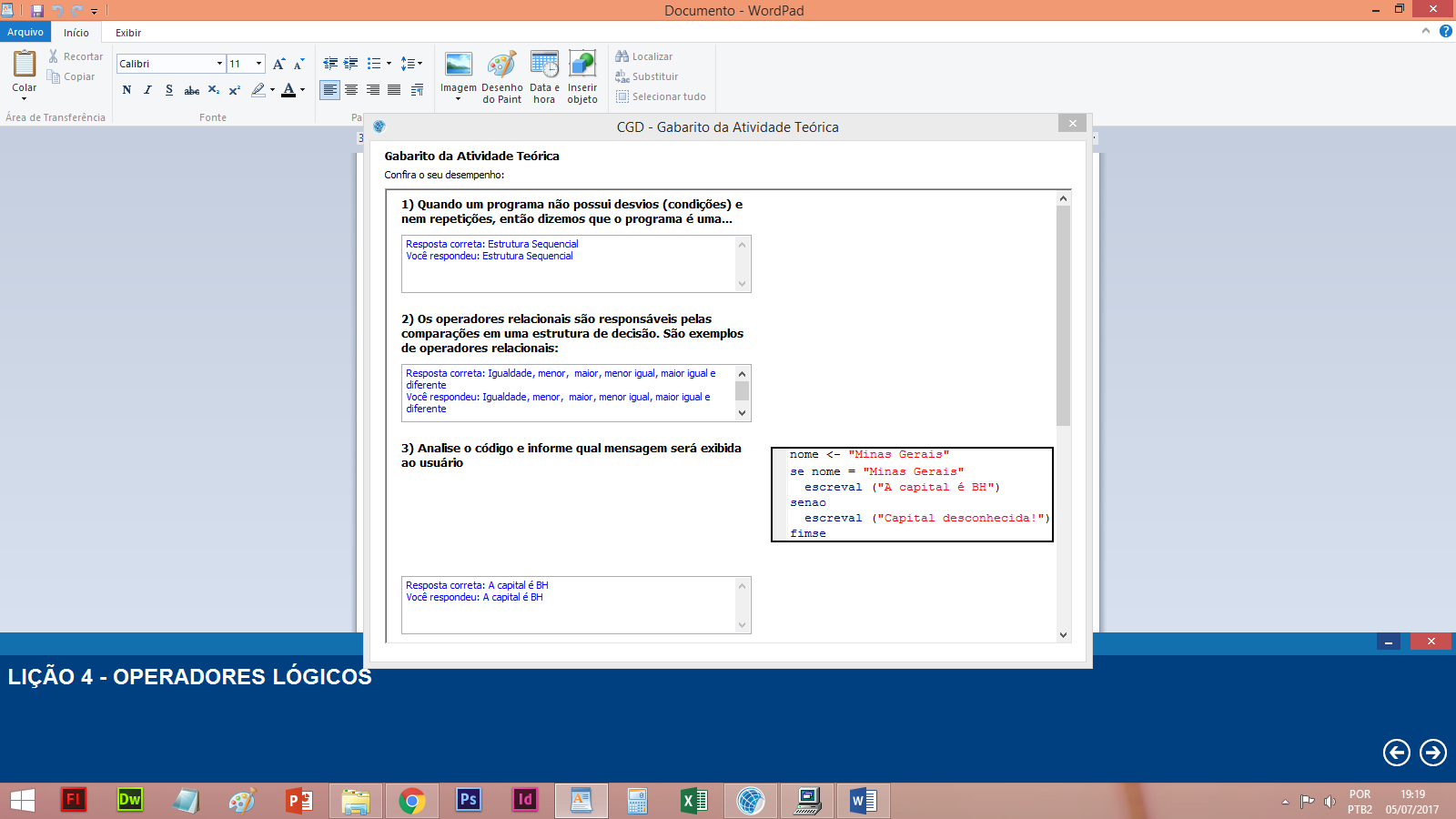
Essa estrutura deve ser utilizada quando o código tiver muitas decisões a serem tomadas, isto é, quando o código oferecer muitas opções de caminhos. Por exemplo, imagine que cada mês informado poderia efetuar um cálculo de desconto diferenciado, então usaríamos o Caso para testar os meses e decidir qual cálculo será apresentado.

Tudo que é feito com a estrutura Caso, também pode ser feita com a condicional SE, porém cada SE tem que ficar aninhado dentro de outra condicional SE, tornando o código muito extenso e de difícil compreensão.

**Sobre a estrutura Caso**

Essa estrutura deve ser utilizada quando o código tiver muitas decisões a serem tomadas, isto é, quando o código oferecer muitas opções de caminhos. Por exemplo, imagine que cada mês informado poderia efetuar um cálculo de desconto diferenciado, então usaríamos o Caso para testar os meses e decidir qual cálculo será apresentado.

Tudo que é feito com a estrutura Caso, também pode ser feita com a condicional SE, porém cada SE tem que ficar aninhado dentro de outra condicional SE, tornando o código muito extenso e de difícil compreensão. 



**OPERADORES LÓGICOS**

**Operações Lógicas**

Vamos iniciar agora o aprendizado sobre operações lógicas, que são operações efetuadas com os valores booleanos (true ou false, [verdadeiro ou falso]). Esse tipo de operação é utilizada nas condicionais SE, e em estruturas de repetição, que ainda não foram aprendidas nesse curso.

"*Marco Antonio Furlan de Souza ... [et al] .Algoritmos e Lógica de Programação. 1. ed. São Paulo: Cengage Learning, 2008. p. 99*".

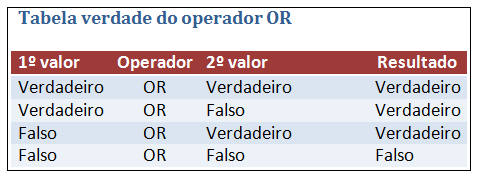
Os operadores lógicos que iremos aprender são os seguintes: **AND**, **OR** e o operador **NOT**.  
Veremos a característica de cada um com exemplos nos algoritmos.

**Operador OR**

Um operador lógico do tipo **OR** realiza o teste simultâneo de duas condições. Através do teste, ele devolve "Verdadeiro" se pelo menos um dos elementos for verdadeiro.

O operador lógico **OR** também pode ser chamado de **OU**.

Observe a tabela verdade:



Uma expressão **OR** é verdadeira se *pelo menos uma condição* for verdadeira.

# Explicação

Dessa vez a cor informada foi "preto".

A condicional testou se era "branco", o resultado foi **FALSO**.

Em seguida, a condicional testou se era "preto", e o resultado foi **VERDADEIRO**.

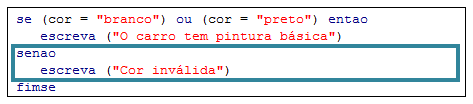
Com isso apresentou novamente a mensagem "O carro tem pintura básica"."

**COM O OPERADOR** OR(OU)-> **A PRIMEIRA CONDIÇÃO** OU  **A SEGUNDA CONDIÇÃO.**

**Explicação**

Nesse teste, a cor informada foi "amarelo". Com isso, a cor não é "branco" e também não é "preto".  
Os 2 testes foram feitos e ambos retornaram FALSO.

Com isso, a condicional SE assumiu o retorno como FALSO.



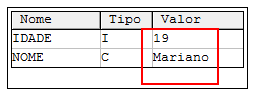
# Explicação

O programa fez o seguinte teste:

C:\Program Files\MetodoCGD\Figuras\logicadeprogramacao\log0545.png

Para satisfazer a condição, isto é, para que ela seja VERDADEIRA -> **idade** tem que ser igual 18 OU **nome** tem que ser igual a "Bruno"

Nos valores informados, ambos as condições são FALSAS:



**Explicação**

O programa fez o seguinte teste:

C:\Program Files\MetodoCGD\Figuras\logicadeprogramacao\log0545.png

Primeiro teste: idade é igual a 18? -> resposta SIM. Com isso já retornou o *primeiro* como **VERDADEIRO**.

Segundo teste: nome é igual a "Bruno"? -> resposta NÃO. Com isso retornou o *segundo* como **FALSO**.

Quando temos um retorno VERDADEIRO e o outro FALSO, o **resultado final é VERDADEIRO**



**Explicação**

O programa fez o seguinte teste:

C:\Program Files\MetodoCGD\Figuras\logicadeprogramacao\log0545.png

Primeiro teste: idade é igual a 18? -> resposta NÃO. A idade é 21, então retornou o *primeiro* como **FALSO**.

Segundo teste: nome é igual a "Bruno"? -> resposta SIM. Com isso retornou o *segundo* como **VERDADEIRO**.

Quando temos um retorno FALSO e o outro VERDADEIRO, o **resultado final é VERDADEIRO**



**Explicação**

O programa fez o seguinte teste:

C:\Program Files\MetodoCGD\Figuras\logicadeprogramacao\log0545.png

Primeiro teste: idade é igual a 18? -> resposta SIM. Retornou o *primeiro* como **VERDADEIRO**.

Segundo teste: nome é igual a "Bruno"? -> resposta SIM. Retornou o *segundo* como **VERDADEIRO**.

Quando temos ambos como VERDADEIRO, o **resultado final é VERDADEIRO**

