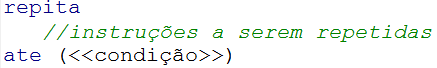
**Introdução**

Agora que você já sabe o que é uma estrutura de repetição e conhece uma delas, é hora de conhecer as outras. Aqui, você saberá quando e como usar as duas estruturas de repetição restantes. Iniciaremos falando sobre a REPITA-ATE, e por fim falaremos da ENQUANTO-FACA. Faça todos os exemplos desse documento no VisuAlg, execute-os com F8 acompanhando o que o documento diz, e com certeza você terá pouquíssimas dúvidas sobre elas. Além disso, tenha em mãos papel e caneta, para anotar as dúvidas que você tiver na leitura desse documento, facilitando assim nossa reunião posterior.

**Estrutura de repetição REPITA-ATE**

A estrutura REPITA-ATE é normalmente usada em casos onde não se sabe a quantidade de vezes que o bloco de instruções deve ser repetido e o *conteúdo do bloco interno deve ser executado pelo menos uma vez*. Segue sua estrutura padrão:



**Obs.: as cores que você vê nas imagens printadas do VisuAlg serão usadas quando nos referirmos a qualquer conteúdo escrito nela (em conjunto com o negrito), então fique atento!**

Veja que as palavras em azul (**repita** e **ate**) são palavras-chave, e devem ser assim escritas para que o VisuAlg reconheça o comando. Você deve lembrar do comentário **//instruções a serem repetidas** do documento anterior. Então isso quer dizer que só precisamos esclarecer a **<<condição>>**.

A **<<condição>>** deve ser uma comparação semelhante às comparações que você usa na estrutura condicional SE, comparando uma variável a um valor ou uma variável com outra. Assim como no SE, é permitido o uso do E e do OU como complementos, caso haja mais de uma condição a ser verificada. No caso do REPITA-ATE, as **//instruções a serem repetidas** serão repetidas até que a condição seja atendida.

O que o REPITA-ATE vai interpretar é:

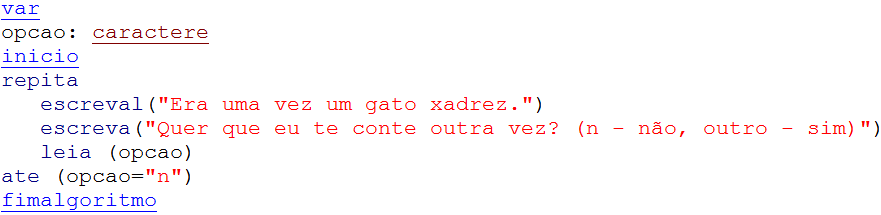
* Quando o **repita** for encontrado, ele compreende que se iniciou uma estrutura de repetição REPITA-ATE;
* As **//instruções a serem repetidas** serão executadas;
* Ao chegar na linha do **ate**, o VisuAlg fará uma verificação:
  + Se a <<condição>> for verdadeira, o VisuAlg encerrará a repetição e a próxima linha a ser executada é a linha após o **ate**;
  + Se a <<condição>> for falsa, ele executará novamente as **//instruções a serem repetidas**.

**Exemplo prático REPITA-ATE**

**ATENÇÃO: para reduzir o tamanho desse documento, os códigos do VisuAlg mostrados se iniciarão após os comentários iniciais e não terão nenhuma linha de comentário, mas isso não é uma prática que pode ser seguida por você!**

**Obs.: nessas simulações, como possuímos alguns comandos “leia”, a simulação de leitura será feita supondo valores digitados pelo usuário, como se estivéssemos mesmo digitando esses valores.**

Veja um algoritmo que conta a história do gato xadrez:

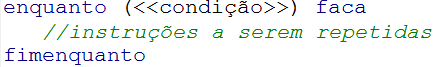


Agora, saiba como o computador vai interpretar o código, a partir de seu **inicio**, na linha 3:

* A linha do **repita** será executada, indicando o início da estrutura de repetição;
* Será mostrada na tela a mensagem “Era uma vez um gato xadrez.”;
* Será mostrada na tela a mensagem “Quer que eu te conte outra vez? (n - não, outro - sim)”;
* Suporemos que o usuário digitou “4” e apertou a tecla ENTER, então esse valor será mostrado na tela, e salvo na variável **opcao**;
* Ao chegar no **ate**, a condição será verificada. Nesse caso, como a condição é falsa (pois o valor da variável **opcao (4)** não é igual a **n**), ele executará novamente as **//instruções a serem repetidas**, então…
* Será mostrada na tela a mensagem “Era uma vez um gato xadrez.”;
* Será mostrada na tela a mensagem “Quer que eu te conte outra vez? (n - não, outro - sim)”;
* Suporemos que o usuário digitou “n” e apertou a tecla ENTER, então esse valor será mostrado na tela, e salvo na variável **opcao**;
* Ao chegar no **ate**, a condição será verificada. Nesse caso, como a condição é verdadeira (pois o valor da variável **opcao (n)** é igual a **n**), a próxima linha a ser executada será a primeira após o **ate**, então…
* **fimalgoritmo**.

**Estrutura de repetição ENQUANTO-FACA**

A estrutura ENQUANTO-FACA é normalmente usada em casos onde não se sabe a quantidade de vezes que o bloco de instruções deve ser repetido *e pode não ser necessário executar o conteúdo do bloco interno*. Segue sua estrutura padrão:



Veja que as palavras em azul (**enquanto, faca** e **fimenquanto**) são palavras-chave, e devem ser assim escritas para que o VisuAlg reconheça o comando. Você sabe o significado de **//instruções a serem repetidas**. Então isso quer dizer que só precisamos esclarecer a **<<condição>>** (e, não, a lógica da **<<condição>>**  do REPITA-ATE não é completamente a mesma no ENQUANTO-FACA, portanto tenha cuidado...).

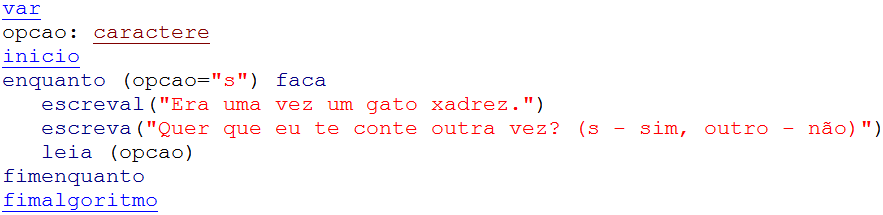
Você já sabe que a **<<condição>>** é uma comparação feita para verificar a necessidade de repetição das **//instruções a serem repetidas**. Mas então o que muda no ENQUANTO-FACA em relação ao REPITA-ATE? O que muda é: enquanto no REPITA-ATE as **//instruções a serem repetidas** serão repetidas até que a condição seja atendida, no ENQUANTO-FACA as **//instruções a serem repetidas** serão repetidas enquanto a condição for atendida.

O que o ENQUANTO-FACA vai interpretar é:

* Quando o **enquanto** for encontrado, o VisuAlg fará uma verificação:
  + Se a <<condição>> for falsa, o VisuAlg ignorará as **//instruções a serem repetidas** e a próxima linha a ser executada será a linha após o **fimenquanto**;
  + Se a <<condição>> for verdadeira, ele executará as **//instruções a serem repetidas**;
* Ao chegar na linha do **fimenquanto**, o VisuAlg voltará para a linha do **enquanto**, e fará novamente a verificação.

**Exemplo prático ENQUANTO-FACA**

Veja o mesmo algoritmo que conta a história do gato xadrez, agora com ENQUANTO-FACA:

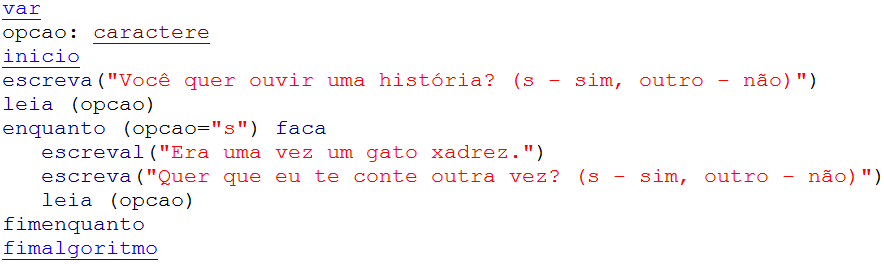


Agora, saiba como o computador vai interpretar o código, a partir de seu **inicio**, na linha 3:

* Ao chegar no **enquanto**, a condição será verificada. Nesse caso, como a condição é falsa (pois o valor da variável **opcao (está vazia)** não é igual a **s**), a próxima linha a ser executada será a primeira após o **fimenquanto**, então…
* **fimalgoritmo**.

Aqui você deve ter reparado e sentido na pele mais uma diferença entre o REPITA-ATE e o ENQUANTO-FACA, que por sinal é visível já em suas estruturas: a **<<condição>>** no REPITA-ATE é verificada *após* a execução das **//instruções a serem repetidas**; e no ENQUANTO-FACA, ela é verificada *antes* de executar as **//instruções a serem repetidas**, ou seja, se a **<<condição>>** não for atendida desde o início, as **//instruções a serem repetidas** nunca serão feitas. Releia o primeiro parágrafo dos tópicos sobre as duas estruturas, focando na parte sublinhada para ficar 100% claro quando usá-las.

Sendo assim, vamos mudar a idéia do algoritmo anterior. Vamos supor que a história só deve ser contada se a pessoa quiser ouvi-la:

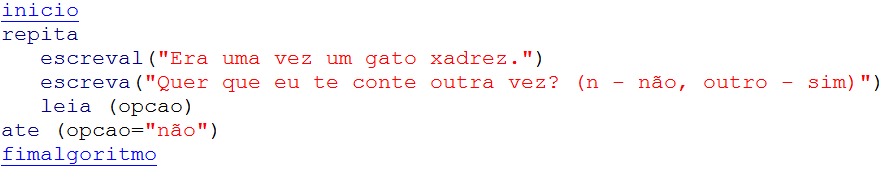


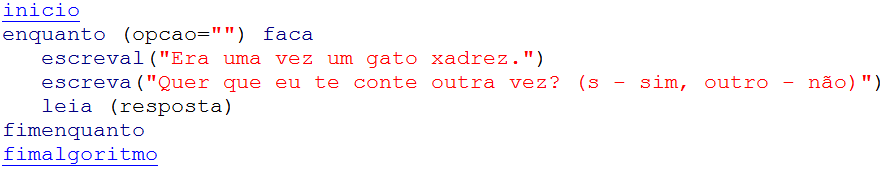
Agora, saiba como o computador vai interpretar o código, a partir de seu **inicio**, na linha 3:

* Será mostrada na tela a mensagem “Você quer ouvir uma história? (s - sim, outro - não)”;
* Suporemos que o usuário digitou “s” e apertou a tecla ENTER, então esse valor será mostrado na tela, e salvo na variável **opcao**;
* Ao chegar no **enquanto**, a condição será verificada. Nesse caso, como a condição é verdadeira (pois o valor da variável **opcao (s)** é igual a **s**), ele entrará na repetição, então…
* Será mostrada na tela a mensagem “Era uma vez um gato xadrez.”;
* Será mostrada na tela a mensagem “Quer que eu te conte outra vez? (s - sim, outro - não)”;
* Suporemos que o usuário digitou “s” e apertou a tecla ENTER, então esse valor será mostrado na tela, e salvo na variável **opcao**;
* Ao chegar no **fimenquanto**, identifica-se o fim das **//instruções a serem repetidas** e volta-se para o início da estrutura de repetição;
* Ao chegar no **enquanto**, a condição será verificada. Nesse caso, como a condição é verdadeira (pois o valor da variável **opcao (s)** é igual a **s**), ele entrará na repetição, então…
* Será mostrada na tela a mensagem “Era uma vez um gato xadrez.”;
* Será mostrada na tela a mensagem “Quer que eu te conte outra vez? (s - sim, outro - não)”;
* Suporemos que o usuário digitou “QUEROOOO!!!” e apertou a tecla ENTER, então esse valor será mostrado na tela, e salvo na variável **opcao**;
* Ao chegar no **fimenquanto**, identifica-se o fim das **//instruções a serem repetidas** e volta-se para o início da estrutura de repetição;
* Ao chegar no **enquanto**, a condição será verificada. Nesse caso, como a condição é falsa (pois o valor da variável **opcao (QUEROOOO!!!)** não é igual a **s**), ele não entrará na repetição, então…
* **fimalgoritmo**.

**Loop infinito nas duas estruturas**

Segue abaixo um loop infinito em cada uma das estruturas de repetição. Cada um gerou loop infinito por motivos diferentes, mas o erro do REPITA-ATE também pode acontecer no ENQUANTO-FACA, e vice-versa. Descubra o que há de errado e tente fazer com que os algoritmos funcionem corretamente.





**ATIVIDADE**

Antes de iniciar a lista de exercícios, você deve ser capaz de preencher a tabela comparativa das estruturas de repetição. **Copie a tabela para um documento do Google Drive para fazer o preenchimento.** Atente-se ao fato de que na tabela comparativa, existe uma coluna para a estrutura para-faça, porém, você viu essa estrutura na lista anterior. A validação dessa tabela será feita juntamente com a lista de exercícios, no entanto, validaremos a tabela sempre antes da lista.

**Tabela comparativa:**

| **Característica** | **PARA-FAÇA** | **REPITA-ATE** | **ENQUANTO-FACA** |
| --- | --- | --- | --- |
| Quando é melhor usá-la? |  |  |  |
| Precisa de contadora para funcionar? |  |  |  |
| Pode causar *loop* infinito? |  |  |  |
| Qual o mínimo de execuções das instruções a serem repetidas possível? |  |  |  |
| O que precisa acontecer para a estrutura parar de executar as instruções a serem repetidas? |  |  |  |