Relatório Técnico do Projecto

CleanSheets

Grupo 2

Elemento
Hugo Miguel Matos Dias 1100592

1 Introdução

1.1 Apresentação

O problema está relacionado com a criação e edição de Macros, que não são mais que um conjunto de instruções que a aplicação CleanSheets terá de executar.

1.2 Requisitos

O utilizador terá de ter acesso na aplicação a um Menu de Macros, onde poderá criar uma Macro com um certo nome e com várias instruções. Poderá também fazer a alteração de Macros anteriormente criadas.

1.3 Objectivos

O utilizador poder alterar e executar Macros. A última instrução é visualizada numa área de "output".

1.4 Dificuldades

As maiores dificuldades do projecto serão a criação de variáveis temporárias, onde as mesmas poderão ter valores ou resultados de fórmulas e que podemos utilizá-las para a utilização de outra fórmula, por exemplo:

$var:=1 →dizer que a $var tem o valor de 1
A1:=sum($var;2) →dizer que a célula A1 irá obter o resultado da soma da variável com 2

1.5 Estrutura do Relatório Técnico

- Introdução;
- Enquadramento;
- Análise;
- Concepção;
- Implementação:
- Conclusão;
- Bibliografia.

2 Enquadramento

2.1 Descrição dos Requisitos

O utilizador deverá ter acesso, ao iniciar a aplicação, a um Menu para criar Macros. O mesmo irá abrir uma janela para a criação e edição de Macros.

Para ser possível criar uma Macro, é necessário inserir um nome para a mesma (nome diferente das macros que já foram criadas, senão a aplicação irá apresentar a seguinte mensagem: "A macro 'xpto' ja esta criada! Pretende substituir o valor da mesma?" e caso a resposta seja positiva irá proceder à edição da Macro em questão caso esteja 100% correcta). Uma Macro poderá conter várias instruções, como por exemplo "A1:=5" ou "A2:=sum(A1:A3)". A aplicação irá verificar se as instruções foram devidamente introduzidas e caso não o tenham sido, o programa irá apresentar uma mensagem relacionada com o problema em questão.
O utilizador poderá também usufruir de variáveis temporárias, onde poderá inserir valores ou fórmulas a uma determinada variável e depois utilizá-la numa instrução futura. Por exemplo:

$var1:=sum(1;2) →Estamos a colocar numa variável chamada de $var1 o valor da soma entre 1 e 2
A1:=$var1 →Estamos a inserir o valor de $var1 na célula A1

2.2 Enquadramento do Projecto

O projecto, no meu ponto de vista, deverá conter o que foi realizado, ou seja o básico, a criação e edição de Macros. Para este projecto, não vejo relevância realizar uma interface interessante porque o que realmente é necessário é o utilizador ter a tarefa mais facilitada e juntar várias instruções numa Macro.

2.3 Funcionalidades

Como já foi descrito anteriormente, as novas funcionalidades da aplicação são: executar macros com várias instruções (desde que as mesmas estejam todas correctas) e edição de macros. O utilizador tem acesso às macros anteriormente criadas e poderá escolher a que pretende fazer alterações. Depois poderá executar novamente a macro e caso esteja 100% correcta, a aplicação faz a alteração da macro e apresenta os resultados na folha de cálculo.
Seja na execução ou na edição, caso a Macro não esteja correcta, a aplicação apresentará uma mensagem certeira relacionado com o problema em questão.
O utilizador também tem acesso ao uso de variáveis temporárias, onde pode inserir valores nas mesmas e utilizá-las numa instrução futura. Também é disponibilizado uma área de "output" para a última instrução inserida.

3 Análise

3.1 Requisitos Funcionais

Diagrama de Casos de Uso

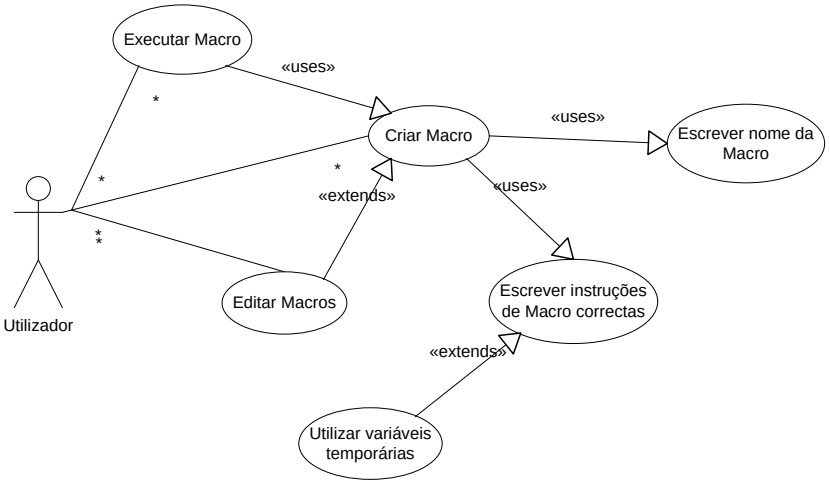
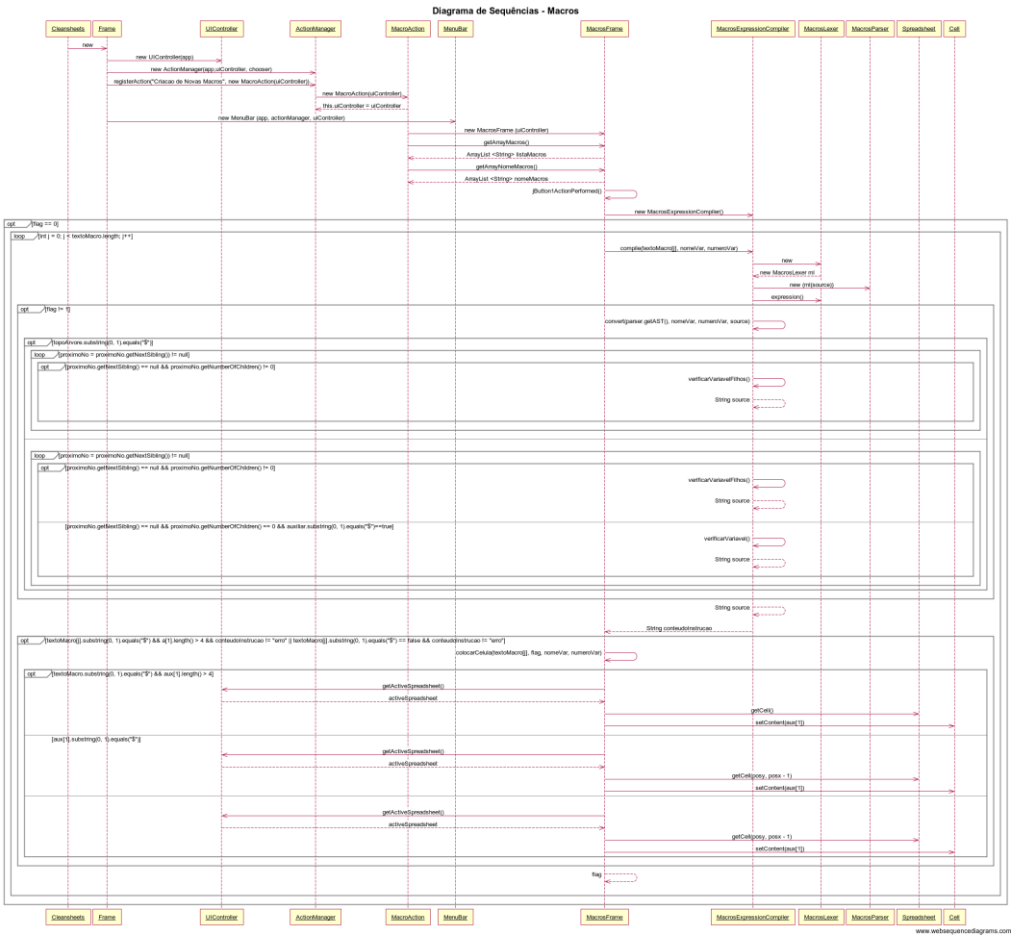
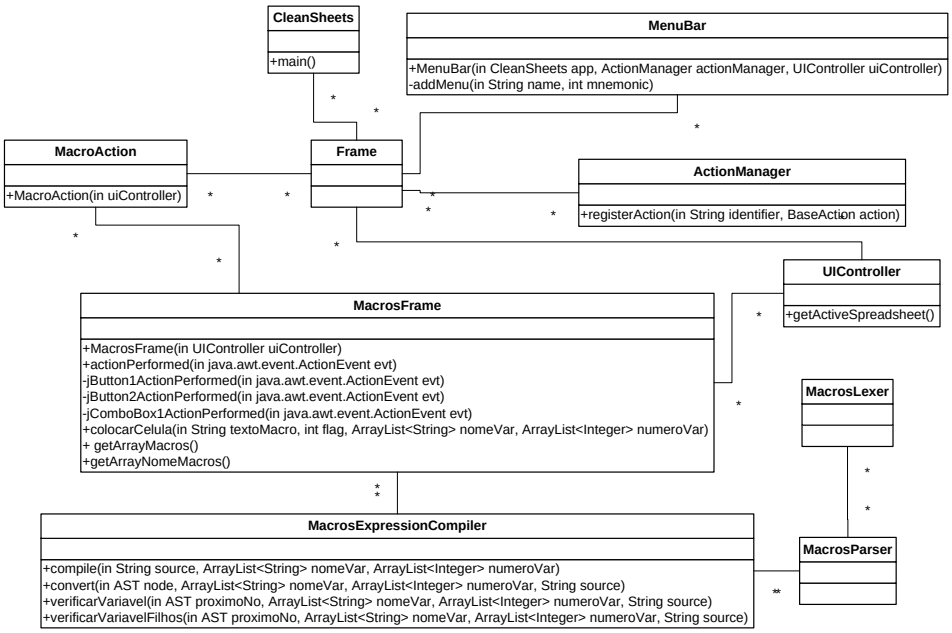


Diagrama de Sequência



Juntamente com este ficheiro, irá a imagem deste diagrama para que o mesmo possa ser visualizado sem problemas.

Diagrama de Classes



3.2 Requisitos Não Funcionais

Não tem requisitos não funcionais.

3.3 Contextualização

O programa já disponibilizava as fórmulas para utilizar nas instruções das macros. Teve de ser criado a possibilidade de uso dessas fórmulas para serem atribuídas a células ou a variáveis. As fórmulas também tiveram de ser ajustadas para ser possível o uso de variáveis nas mesmas.

3.4 Planeamento

- Estudo da nova iteração, o que é pedido exactamente;
- Criação dos novos diagramas (seguência, classes, use-case);
- Criação de uma nova gramática para as macros;
- Criação de variáveis temporárias e respectiva utilização;
- Criação de uma área de “output” para a última instrução da macro;

4 Concepção

4.1 Especificação

Os mesmos da análise.

4.2 Aspectos técnicos da solução

Para esta solução, foi necessário a criação de uma nova gramática para as macros (algo que me tinha proposto a fazer na última iteração) onde criei um novo ficheiro .g com a especificação da nova gramática que irá verificar cada instrução de uma macro e que deve ser da seguinte maneira:

- Deverá ter uma referência a uma célula (A112 por exemplo) ou a uma variável ($var1 por exemplo), seguido de um ":=" seguido de uma fórmula ou valor em que a mesma também poderá ter uma variável (por exemplo A1:= $var1 ou A1:=sum($var1;2)).

Para a criação e utilização das variáveis temporárias foi necessário percorrer a árvore gerada através da nossa gramática, onde verifico todas as hipóteses possíveis, eis o exemplo de algumas:

-$var:=1
-A1:=$var
-A1:=sum($var;2)

Na primeira linha, o programa atribuí o valor 1 à variável $var, através do armazenamento em ArrayLists, uma para o nome da variável, outro para o seu valor.

Na segunda linha, o programa irá atribuir o valor de $var à célula A1, para tal deverá percorrer o ArrayList e verificar se a variável já tinha sido criada anteriormente, caso não tenha o valor de A1 será 0.

Na terceita linha é a mesma lógica que na segunda, só que agora teremos de fazer o cálculo da fórmula primeiro e só depois acrescentar à célula A1.

Para a criação da área de "output" apenas foi necessário inserir numa área de texto o valor resultante da última instrução.

4.3 Testes

4.3.1 Testes unitários

4.3.2 Casos de teste

Semana 1
(28 de Maio a 3 de Junho)

Nome do caso de teste: *Criação de uma Macro*
Casos de uso relacionados:

Objectivo	*O utilizador inserir instruções e executar uma Macro*
Pré-requisitos	*É necessário colocar um nome na Macro* *A Macro deverá estar totalmente certa*
Dados de teste	*Por exemplo, inserir a seguinte linha da janela da criação de macros: "A1:=sum(A2:A3)"*
Passos	*1. Ir ao menu Macros* *2. Escolher a opção "Criação de Novas Macros"* *3. Introduzir um nome à Macro* *4. Inserir a instrução da Macro pretendida* *5. Clicar no botão "executar Macro"*
Notas e Questões	*Para criar uma Macro deverá ter sido introduzido um nome para a mesma e pelo menos uma instrução e que deverá estar 100% correcta.*

Comentário [Alexandre1]: Exemplo de descrição de um caso de teste. A substituir pelos casos de teste do projecto.

Resultados

#Execução	Dados	Resultados	Passou?	Observações
#1	$var1:=5	A variável var1 ficou com o valor 5	Sim	A variável "$var1" fica com o valor de 5
#2	A1=$var1	A célula A1 deveria ficar com o valor de 5	Não	Não foi introduzida a instrução correcta. Deve ser do formato célula/variável + ":=" + fórmula/formula com variável
#3	A1:=$var		Sim/Não	Passa caso a variável "$var1" tenha algum valor, caso não tenha é atribuído 0
#4	A:=1		Não	O destino da fórmula deverá ser sempre constituído por uma letra e pelo menos um número
#5	A1:=som(A1:A3)		Não	A fórmula está mal escrita

5 Implementação

Não existe nenhuma dependência de ficheiros de configuração, o utilizador poderá executar a aplicação normalmente. O código principal, o “núcleo” da criação da janela e respectiva execução e edição de macros está responsável a classe MacrosFrame. A classe MacroAction está responsável de criar uma nova MacrosFrame, enviado um uiController, crucial para o funcionamento da classe MacrosFrame. Esse mesmo uiController é enviado à classe MacroAction pela classe Frame, onde a mesma utiliza a seguinte linha de código para criar o submenu e enviar o tal uiController à MacroAction: “actionManager.registerAction("Criacao de Novas Macros", new MacroAction(uiController));”

Foi criada uma classe nova chamada MacrosExpressionCompiler que serve para percorrer e verificar a árvore de Parser gerada pela nossa gramática. As classes MacrosParser, MacrosLexer, MacrosParserTokenTypes foram geradas automaticamente.

Para esta iteração, o código da classe MacrosFrame foi mudado quase na totalidade para ser adaptado à nova lógica do programa ou seja, à nova gramática.

6 Conclusão

Na minha opinião foi realizado com sucesso o que foi pedido nesta iteração, existiam várias maneiras de a realizar, eu optei pela maneira que me tinha proposto na última iteração, ou seja, através da criação de uma nova gramática. Deu bastante trabalho a implementá-la mas fico contente porque está a dar na perfeição.

Em relação à criação de variáveis temporárias, também existiam várias maneiras de resolver a situação, eu optei por utilizar a gramática que realizei e a árvore que ele gerou para resolver o problema.

7 Bibliografia

http://regexpal.com/

http://www.java-tips.org/java-se-tips/javax.swing/how-to-create-menu-bar.html

http://www.java-tips.org/java-se-tips/java.awt.event/what-types-of-events-exist-in-swing-4.html

http://tips4java.wordpress.com/2008/11/11/sorted-combo-box-model/

http://docs.oracle.com/javase/tutorial/uiswing/components/combobox.html

http://docs.oracle.com/javase/1.4.2/docs/api/org/w3c/dom/Node.html