Relatório Técnico do Projecto

CleanSheets

Grupo 2

Elemento
1100554 – Bruno Cunha

1 Introdução

1.1 Apresentação

O problema proposto na 2ª iteração na área Core consiste em capacitar a aplicação de reconhecer novas instruções, como sequências de fórmulas ou atribuição directa de valores a células.

1.2 Requisitos

Permitir escrever fórmulas nos mais diversos contextos, especificamente sequências de fórmulas numa só instrução e atribuições directas de valores.

1.3 Objectivos

O objectivo principal é que a aplicação seja capaz de interpretar as novas instruções iniciadas pelo caractere “#”.

1.4 Dificuldades

A principal dificuldade foi a ambientação com o sistema ANTLR, utilizado previamente no projecto. Foi necessária bastante pesquisa, pois foi algo “novo”. Esta iteração é substancialmente mais difícil que a anterior, pois envolve uma maior manipulação e integração do sistema ANTLR, mas torna-se, ao mesmo tempo, muito mais interessante.

1.5 Estrutura do Relatório Técnico

1-Introdução
2- Enquadramento
3-Análise
4-Concepção
5-Conclusão
6-Bibliografia

2 Enquadramento

2.1 Descrição dos Requisitos

Permitir escrever fórmulas nos mais diversos contextos, especificamente sequências de fórmulas numa só instrução e atribuições directas de valores*:*

O programa deverá reconhecer um novo tipo de expressões, e deverá também ser capaz de encadear várias expressões numa só.

2.2 Enquadramento do Projecto

O requisito principal do projecto, nesta iteração, é o reconhecimento das expressões pedidas, que serão inseridas na gramática Portuguesa. É algo novo à aplicação, e que será exclusivo da gramática Portuguesa.

2.3 Funcionalidades

A funcionalidade requisitada para “melhorar” a aplicação foi o reconhecimento de expressões e, após este, o reconhecimento de sequências dessas expressões, na língua Portuguesa – a gramática criada na iteração anterior.

3 Análise

3.1 Requisitos Funcionais

Diagrama Sequência:

(Inclui apenas os casos de uso desta iteração)

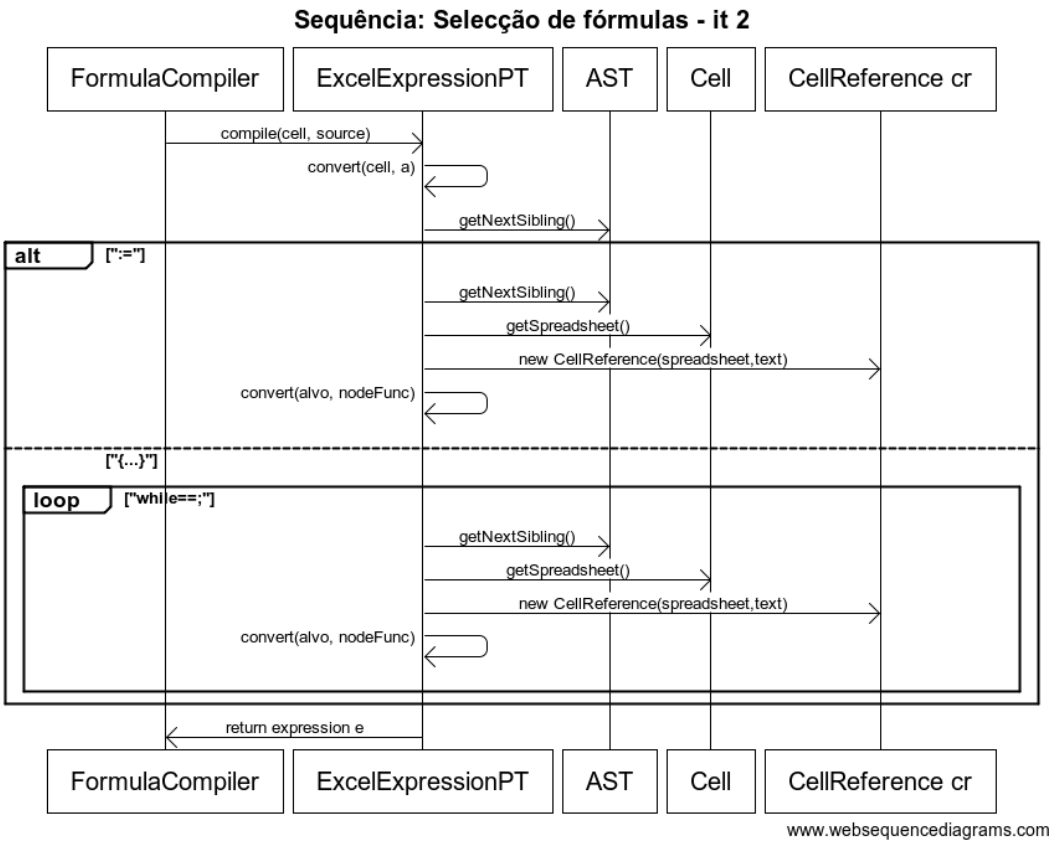


Diagrama Casos de Uso:

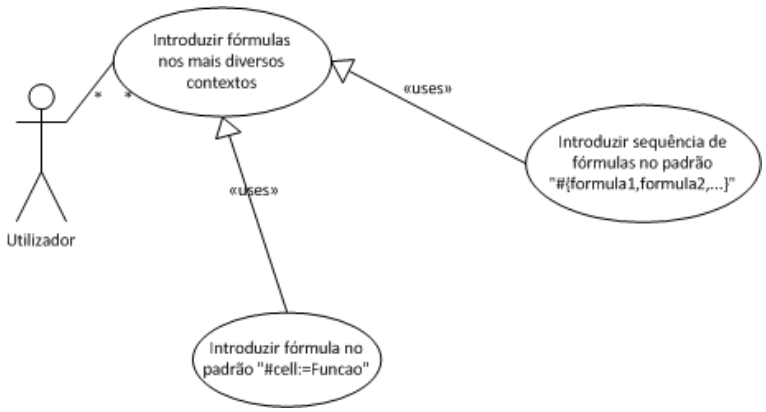
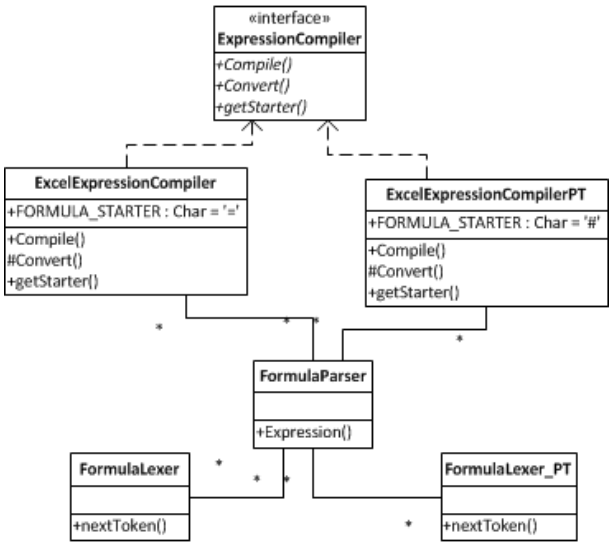


Diagrama Classes:



3.2 Requisitos Não Funcionais

- A solução deverá ser multiplataforma, incluindo, no mínimo, suporte para um sistema operativo da família Windows e para uma distribuição de GNU/Linux;
- A solução também não deverá depender em particular de nenhum SGBD Relacional

3.3 Contextualização

O requisito principal do projecto, nesta iteração, é o reconhecimento das expressões pedidas, que serão inseridas na gramática Portuguesa. Também devem ser reconhecidas sequências destas-

3.4 Planeamento

Todo o planeamento pode ser consultado no ficheiro integrado no repositório.

4 Concepção

4.1 Especificação

Diagrama Sequência:

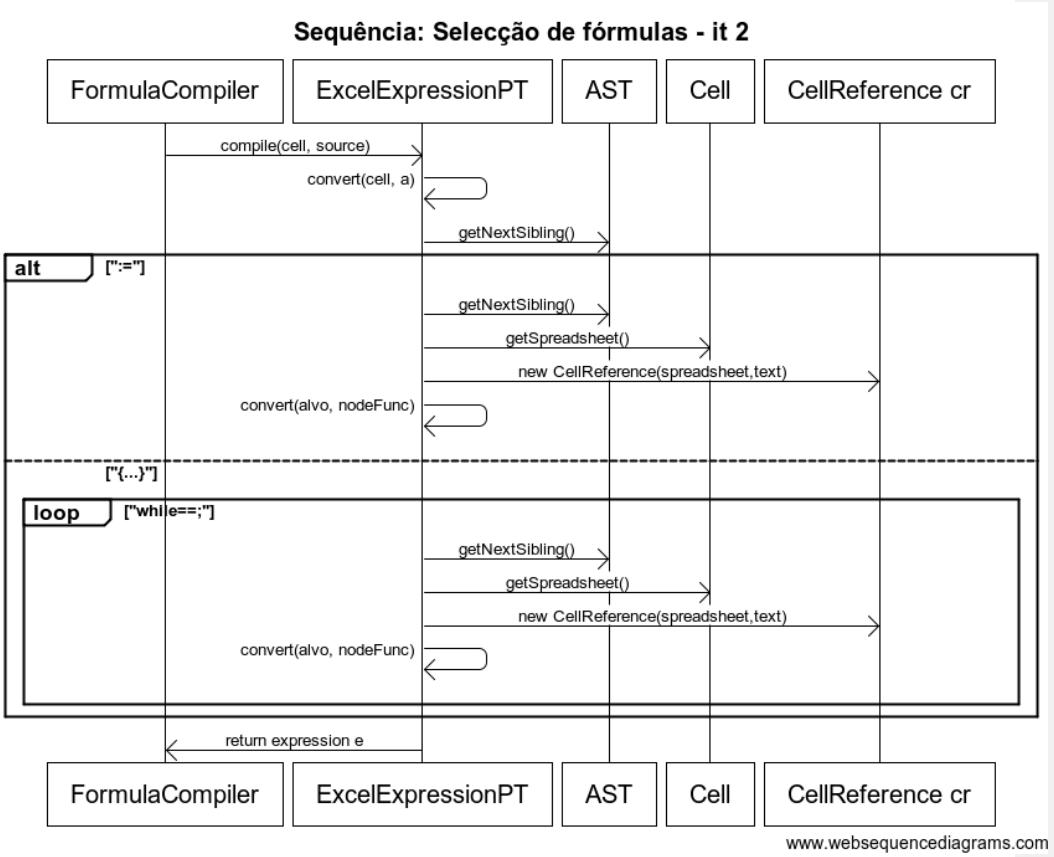


Diagrama Casos de Uso:

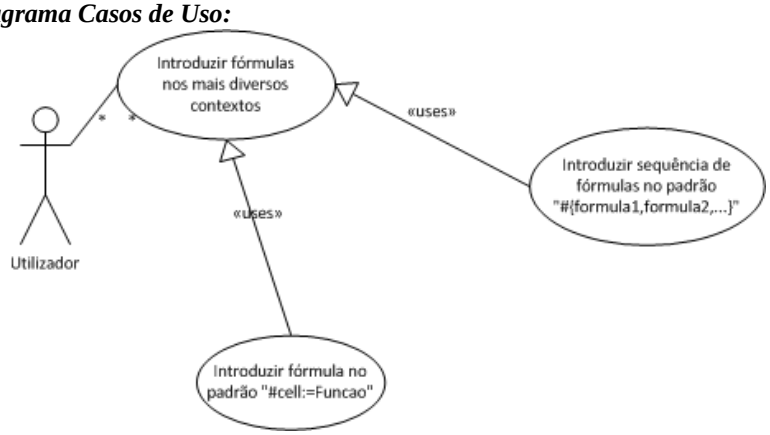
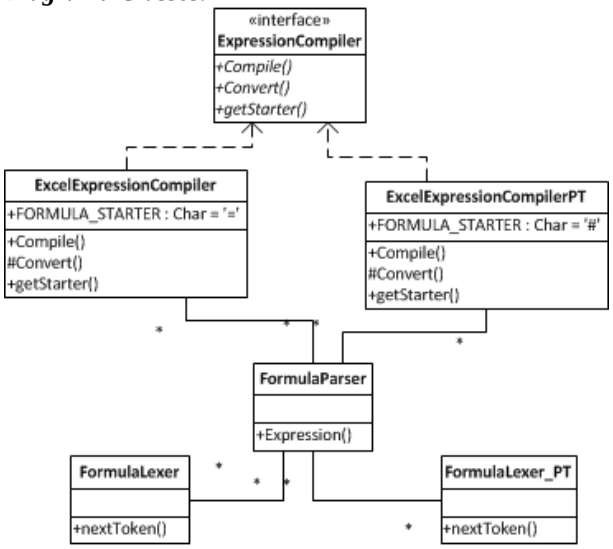


Diagrama Classes:



4.2 Aspectos técnicos da solução

Foi utilizada uma interface comum às gramáticas. É utilizado o ANTLR para gerar *parsers*. Neste caso o ANTLR está também a gerar árvore sintáctica resultante do "parsing" (AST). A AST é percorrida (método "convert") e os seus nodos são convertidos em instâncias de "Expression" que são usadas para executar a expressão. Para analisar as expressões relativas à iteração 2, são analisados os "nodes" gerados pelo sistema ANTLR, ou seja, é analisada a árvore léxica gerada pela expressão introduzida.

4.3 Testes

4.3.1 Testes unitários

```
public class ExcelExpressionCompilersTest {
   Workbook wb = new Workbook(2);
   Spreadsheet s = wb.getSpreadsheet(0);
   ExcelExpressionCompilerPT instancePT = new ExcelExpressionCompilerPT();
   ExcelExpressionCompiler instance = new ExcelExpressionCompiler();



   @Before
   public void setUp() {
      Language.getInstance();
      s.setTitle("titulo");
   }

   @Test
   public void testGetStarterPT() {
      System.out.println("getStarterPT");
      char expResult = '#';
      char result = instancePT.getStarter();
      assertEquals(expResult, result);
   }

   @Test
   public void testGetStarter() {
      System.out.println("getStarter");
      char expResult = '=';
      char result = instance.getStarter();
      assertEquals(expResult, result);
   }

   /**
    * Test of compile method
    */
   @Test
```

```
    public void testCompile() throws Exception {
        System.out.println("compile");
        String source = "#Media(2;4;6)";
        Cell cell = s.getCell(1, 1);
        Formula f = FormulaCompiler.getInstance().compile(cell, source);
        assertEquals(true, f.toString().length()>0);
    }
}
```

4.3.2 Casos de teste

Semana 1
(28 de Maio a 3 de Junho)

Nome do caso de teste: *Base de Dados*

Comentário [Alexandre1]:

Casos de uso relacionados:

Objectivo	*O utilizador deverá conseguir graver informação na Base de dados dos seguintes SGBD MySQL, Postgres e SQLserver.*
Pré-requisitos	*User is not already logged in.* *User testuser exists, and account is in good standing.*
Dados de teste	*Nome da base de dados* *Endereço e porta onde a bd está instalada* *Username e pass para aceder à base de dados*
Passos	1. *Escolher o SGBD dos 3 disponíveis* 2. *Escolher a área das Células* 3. *Indicar os dados para a conexão* 4. *Indicar o nome da tabela*
Notas e Questões	*O utilizador deverá ter os servidores do SGBD para testar no seu computador e criar os respectivos utilizadores + pass de maneira a conseguir aceder.*

Resultados

#Execução	Dados	Resultados	Passou?	Observações
#1	Postgres A1-B2 Dados de conexão	Maria Joana 150 200	Sim, gravou com sucesso	
#2	Postgres A1-B2 Dados de conexão	Maria Joana 150 200	Sim, gravou com sucesso	

#3	SQLserver A1-B2 Dados de conexão	Maria Joana 150 200	Sim, gravou com sucesso	

5 Implementação

- Existem dependências de 3 ficheiros de configuração, referentes às 2 gramáticas e aos seus compiladores de fórmulas. São também utilizados os scripts para criar “dinamicamente” os “lexers” das 2 gramáticas.

6 Conclusão

Esta iteração foi concluída com sucesso, tendo sido cumpridos todos os requisitos. Ficou totalmente funcional o reconhecimento, não só das expressões no formato pretendido, mas também de sequências de 2 ou mais expressões encadeadas.

7 Bibliografia:

http://www.antlr.org/

http://csheets.sourceforge.net/api/