

Relatório Técnico do Projecto

CleanSheets

Grupo 2

Elemento

1100554 – Bruno Cunha

1 Introdução

1.1 Apresentação

O problema proposto na 1ª iteração na área Core consiste em criar uma nova gramática que reconheça fórmulas na língua Portuguesa.

1.2 Requisitos

Permitir escrever fórmulas nos formatos mais comuns para os novos utilizadores, especificamente em Português.

1.3 Objectivos

O objectivo principal é que a aplicação reconheça fórmulas na língua Portuguesa, iniciadas pela caractere “#”.

1.4 Dificuldades

A principal dificuldade foi a ambientação ao código já existente. Foi necessária alguma precaução para manter a nova gramática com uma estrutura “igual” à já existente.

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 formatos mais comuns para os novos utilizadores, especificamente em Português:

O programa deverá reconhecer fórmulas introduzidas em Português, utilizando uma nova gramática.

2.2 Enquadramento do Projecto

O requisito principal do projecto, no momento, é o reconhecimento das fórmulas numa nova gramática, especificamente em Português. O projecto já continha uma gramática em Inglês, que foi “tomada” como base para a adaptação para a nova língua. Só foram acrescentadas fórmulas que já existissem na língua Inglesa, ou seja, não foram adicionadas novas fórmulas.

2.3 Funcionalidades

A funcionalidade requisitada para “melhorar” a aplicação foi o reconhecimento de fórmulas na língua Portuguesa – nova gramática.

3 Análise

3.1 Requisitos Funcionais

Diagrama Sequência:

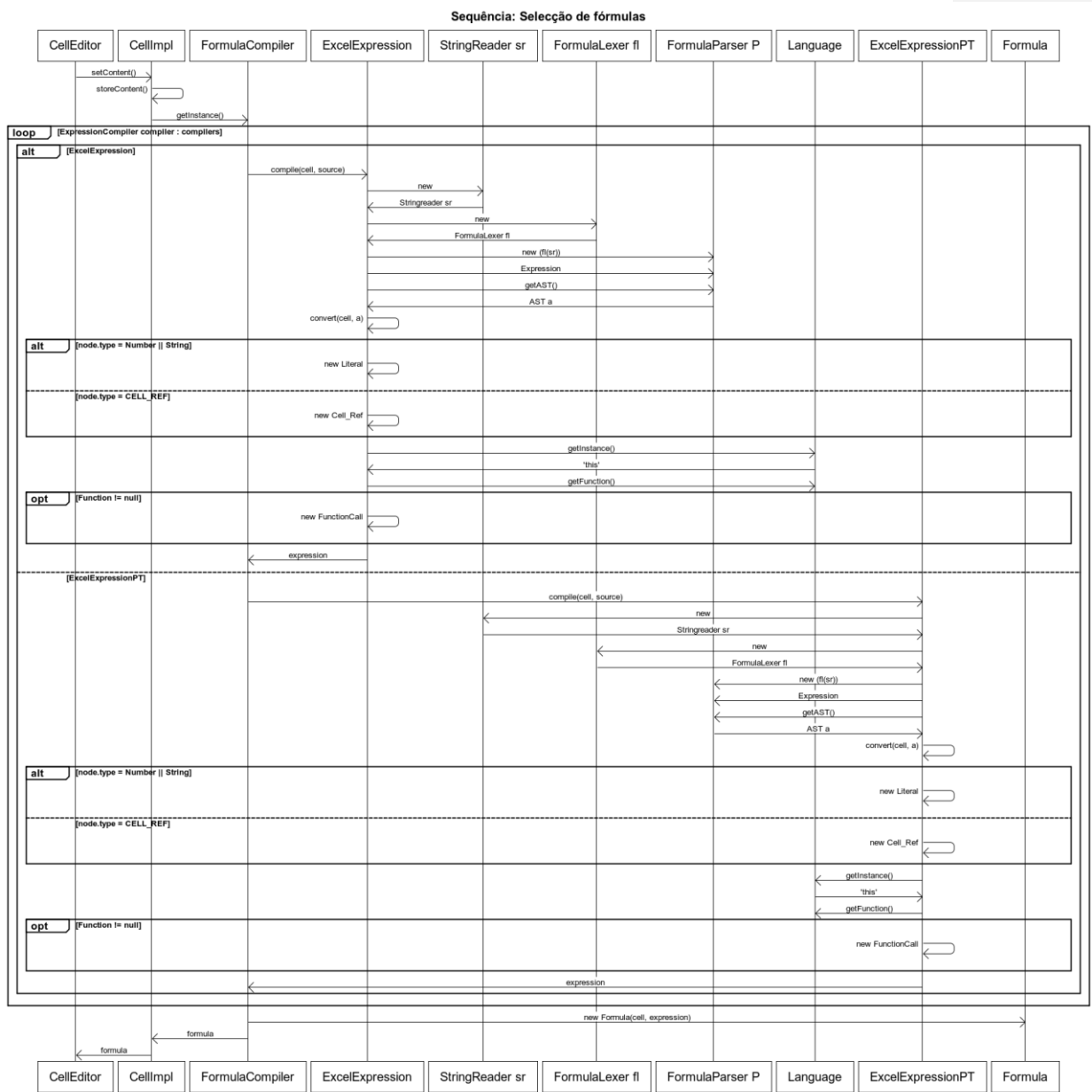


Diagrama Casos de Uso:

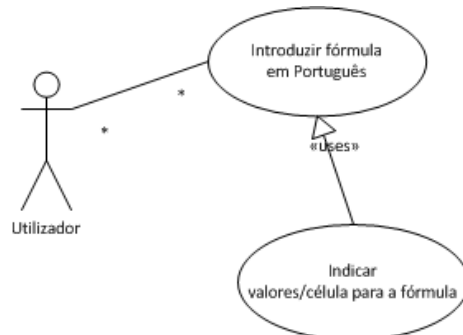
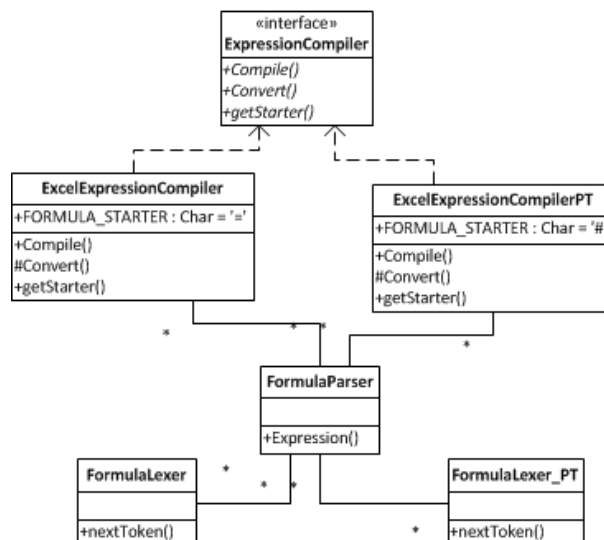


Diagrama Classes:



3.2 Requisitos Não Funcionais

Não disponíveis nesta iteração.

3.3 Contextualização

O requisito principal do projecto, no momento, é o reconhecimento das fórmulas numa nova gramática, especificamente em Português. O projecto já continha uma gramática em Inglês, que foi “tomada” como base para a adaptação para a nova língua. Só foram acrescentadas fórmulas que já existissem na língua Inglesa, ou seja, não foram adicionadas novas fórmulas.

3.4 Planeamento

Apresentar o planeamento do trabalho com base nas técnicas do módulo de gestão de projectos

4 Concepção

4.1 Especificação

Diagrama Sequência:

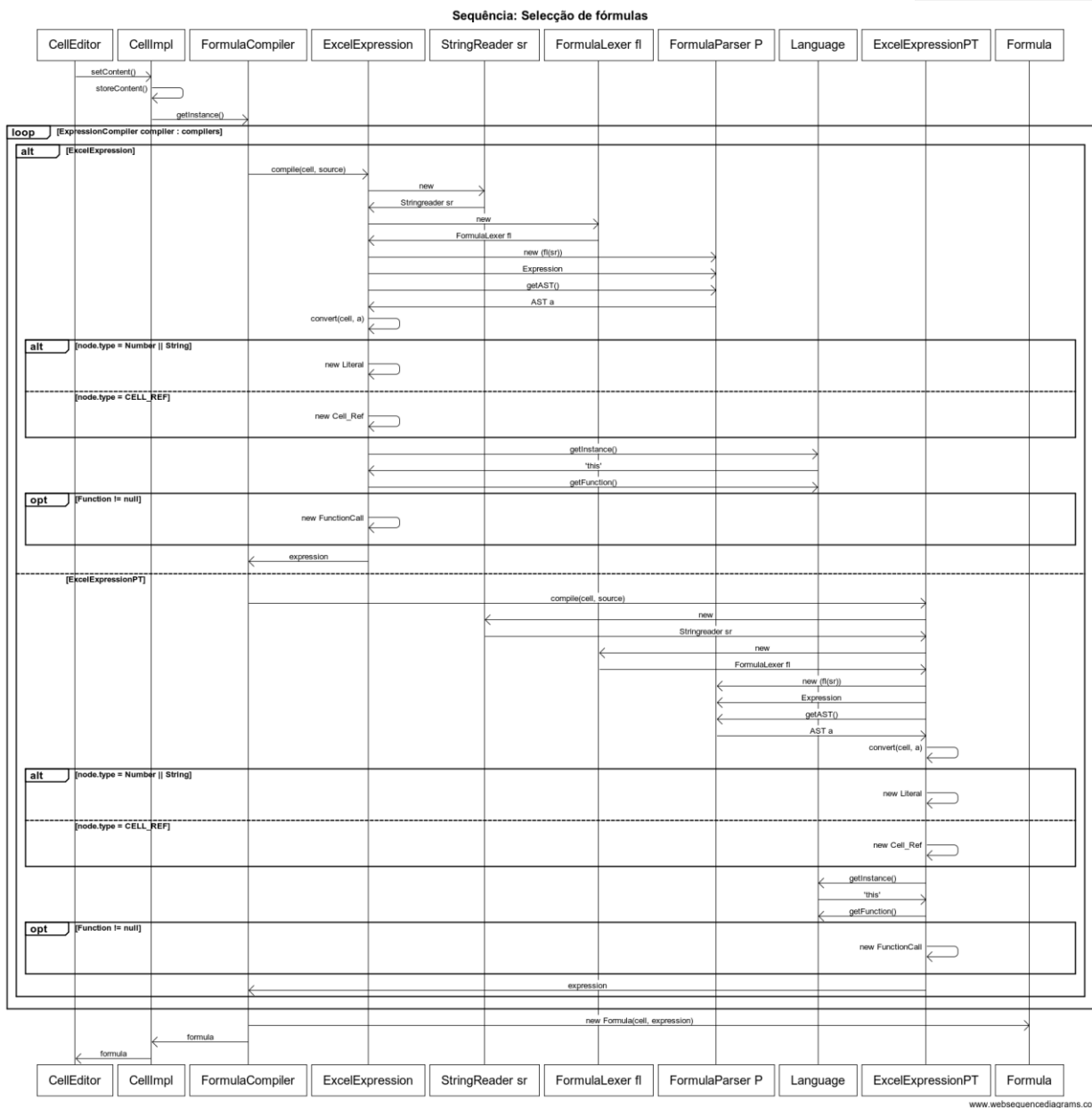


Diagrama Casos de Uso:

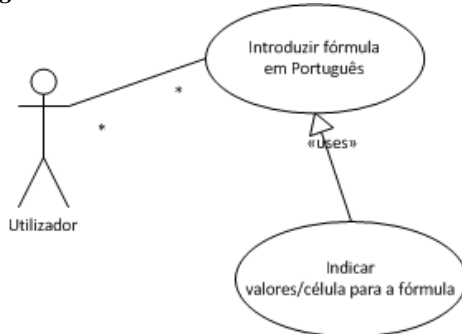
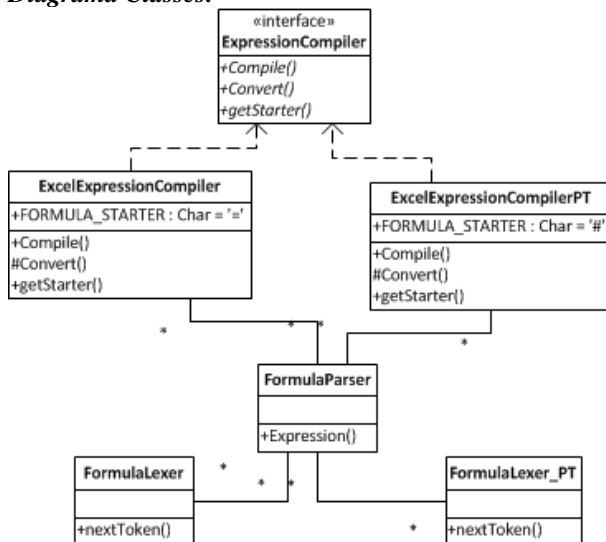


Diagrama Classes:



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.

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*

Casos de uso relacionados:

Objectivo	<i>O utilizador deverá conseguir graver informação na Base de dados dos seguintes SGBD MySQL, Postgres e SQLserver.</i>
Pré-requisitos	<i>User is not already logged in. User testuser exists, and account is in good standing.</i>
Dados de teste	<i>Nome da base de dados Endereço e porta onde a bd está instalada Username e pass para aceder à base de dados</i>
Passos	<i>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</i>
Notas e Questões	<i>O utilizador deverá ter os servidores do SGBD para testar no seu computador e criar os respectivos utilizadores + pass de maneira a conseguir aceder.</i>

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	SQLserv er A1-B2 Dados de conexão	Maria Joana 150 200	Sim, gravou com sucesso	

Comentário [Alexandre1]:

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 a nova gramática, sendo o programa capaz de reconhecer/distinguir fórmulas em Português e Inglês.

7 Bibliografia:

<http://wwwantlr.org/>

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