FERRAMENTA PARA CRIAÇÃO DE BASES DE CONHECIMENTO NA FORMA DE ONTOLOGIA OWL A PARTIR DE DADOS NÃO ESTRUTURADOS

Aluno: Allan Renato Sabino

Orientador: Roberto Heinzle

FURB
UNIVERSIDADE DE BLUMENAU

Roteiro

- Introdução;
- Objetivos;
- Fundamentação teórica;
- Trabalhos correlatos;
- Requisitos;
- Especificação;
- Implementação;
- Operacionalidade da implementação;
- Resultados e discussões;
- Conclusões e sugestões.

Introdução

- O avanço das tecnologias promoveu significativo aumento no armazenamento de informações nas organizações;

- Repositórios de dados tornaram-se maiores.

- Dados não são insumos suficientes ao gestor para tomada de decisão;

- A transformação de dados e informações em conhecimento qualifica a tomada de decisão.

- Ferramentas que automatizam esse processo tornaram-se essenciais;

- Conhecimento formalizado através de ontologia OWL.

Objetivos

- Este trabalho tem como objetivo disponibilizar uma ferramenta para a criação de bases de conhecimento na forma de uma ontologia OWL a partir de textos não estruturados;

- Os objetivos específicos do trabalho são:
  - possibilitar a visualização dos documentos que compõem a base textual;
  - disponibilizar a estrutura morfológica da base textual;
  - apresentar as estruturas ontológicas a partir do conhecimento descoberto.

Fundamentação Teórica

Era da informação e do conhecimento

- Transição para o novo milênio foi marcada por mudanças rápidas e de grande impacto na sociedade;
- Dado;
- Informação;
- Conhecimento.

Ontologia

- Ontologia:
  - *ontos* (ser);
  - *logos* (conhecimento sobre).
- Trata da natureza e organização do ser;
- Representação de um vocabulário relacionado a um certo domínio.

Linguagem OWL

- Integra tecnologias recomendadas pela W3C;
- Baseada em XML e RDF;
- Três sub-linguagens:
  - Lite;
  - DL;
  - Full.

- Quatro mecanismos para formalizar conhecimento:
  - Classe;
  - Relacionamento;
  - Objeto;
  - Axiomas;

Descoberta de conhecimento

- DCDB:
  - Mineração de dados.
- DCT:
  - Sumarização;
  - Agrupamento de documentos;
  - Mineração de texto

Mineração de texto

- Cinco etapas:
  - Coleta;
  - Pré-processamento;
  - Indexação;
  - Mineração;
  - Análise.

Trabalhos Correlatos

- **Uma abordagem semi-automática para a identificação de estruturas ontológicas a partir de textos na língua portuguesa do Brasil;**

- **Um processo semi-automático para o povoamento de ontologias a partir de fontes textuais**

- **Um sistema para extração de informação em referências bibliográficas baseado em aprendizagem de máquina.**

Características	Baségio (2007)	Faria e Girardi (2010)	Silva (2004)
Identificar estruturas ontológicas	X	-	-
Popular ontologia OWL	-	X	-
Criar Ontologia OWL	X	-	-
AM	-	-	X
Remover *stopwords*	X	-	X
Lematizar palavras	X	-	X
Realizar indexação	X	-	X
Realizar mineração	X	X	X

Requisitos

- Os RF da ferramenta são:
  - RF 01: A ferramenta deverá permitir a manutenção de bases textuais;
  - RF 02: A ferramenta deverá permitir a definição da base textual de trabalho.

– RF 03: A ferramenta deverá realizar a coleta de documentos com formato txt;

– RF 04: A ferramenta deverá popular a base textual de trabalho com os documentos coletados.

- RF 05: A ferramenta deverá exibir a estrutura da base textual;

- RF 06: A ferramenta deverá exibir a base textual de trabalho anotada morfologicamente.

– RF 07: A ferramenta deverá exibir a estrutura do arquivo índice invertido criado a partir da base textual de trabalho;

– RF 08: A ferramenta deverá consultar documentos a partir de uma palavra fazendo uso do arquivo de índice invertido criado.

- RF 09: A ferramenta deverá exibir a base textual de trabalho sem as *stopwords*;
- RF 10: A ferramenta deverá exibir a base textual lematizada;
- RF 11: A ferramenta deverá descobrir conhecimento da base textual de trabalho.

– RF 12: A ferramenta deverá exibir as estruturas ontológicas (classes e seus relacionamentos) a partir do conhecimento extraído;

– RF 13: A ferramenta deverá criar uma base de conhecimento com o conhecimento extraído.

- Os RNF da ferramenta são:
  - RNF 01: A ferramenta Enterprise Architect (EA) na versão 7.5 deverá ser utilizada para realizar a especificação;
  - RNF 02: A ferramenta deverá ser implementada utilizando a linguagem de programação Java na versão 7.0.

– RNF 03: A ferramenta deverá ser implementada utilizando o ambiente de desenvolvimento Eclipse na versão Juno;

– RNF 04: A ferramenta deverá utilizar ontologias OWL como formalismo de RC utilizado pela base de conhecimento.

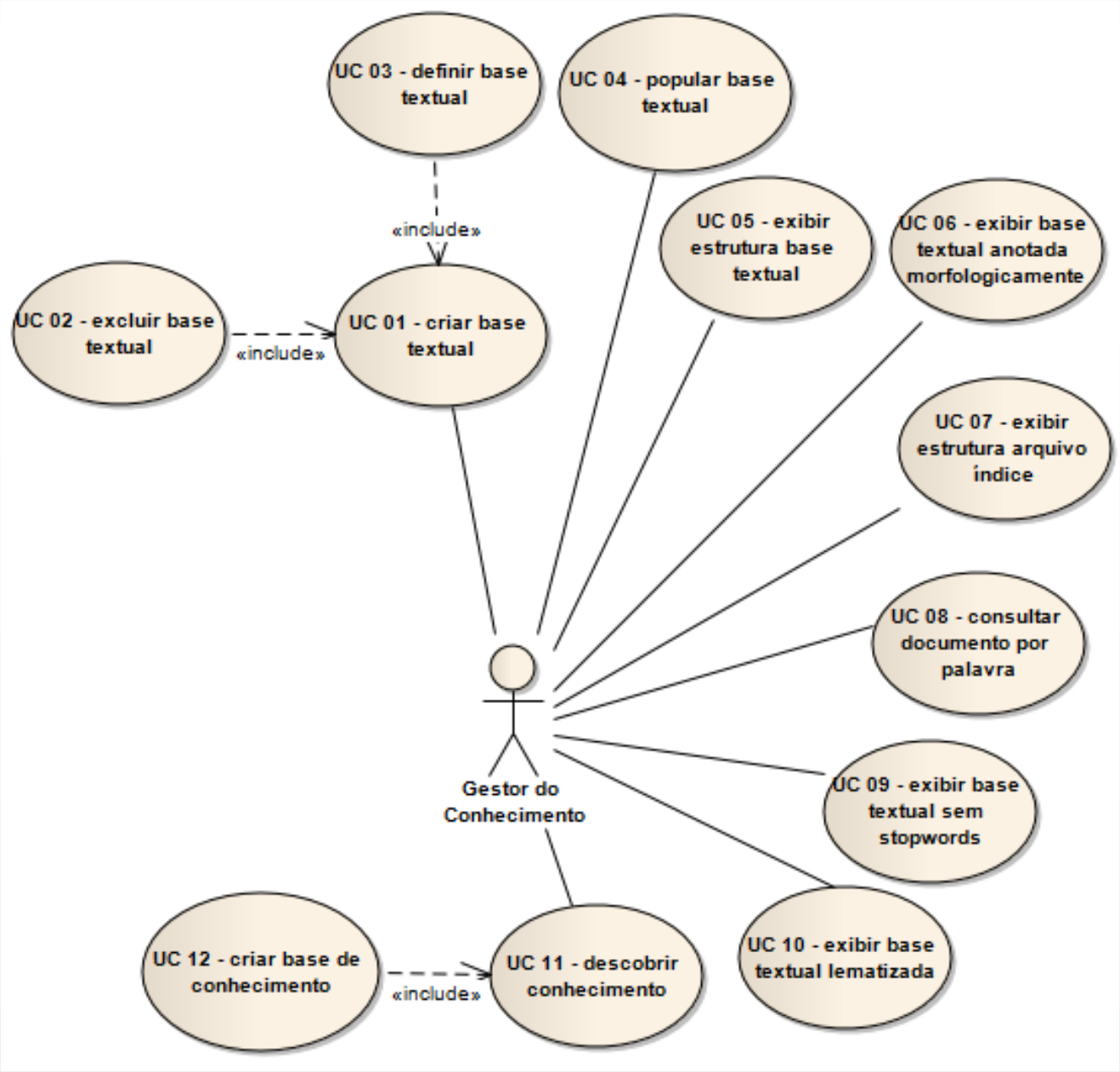
– RNF 05: A ferramenta deverá utilizar a biblioteca Cogroo (USP, 2011) na versão 4.0 para realizar a análise morfológica da base textual de trabalho;

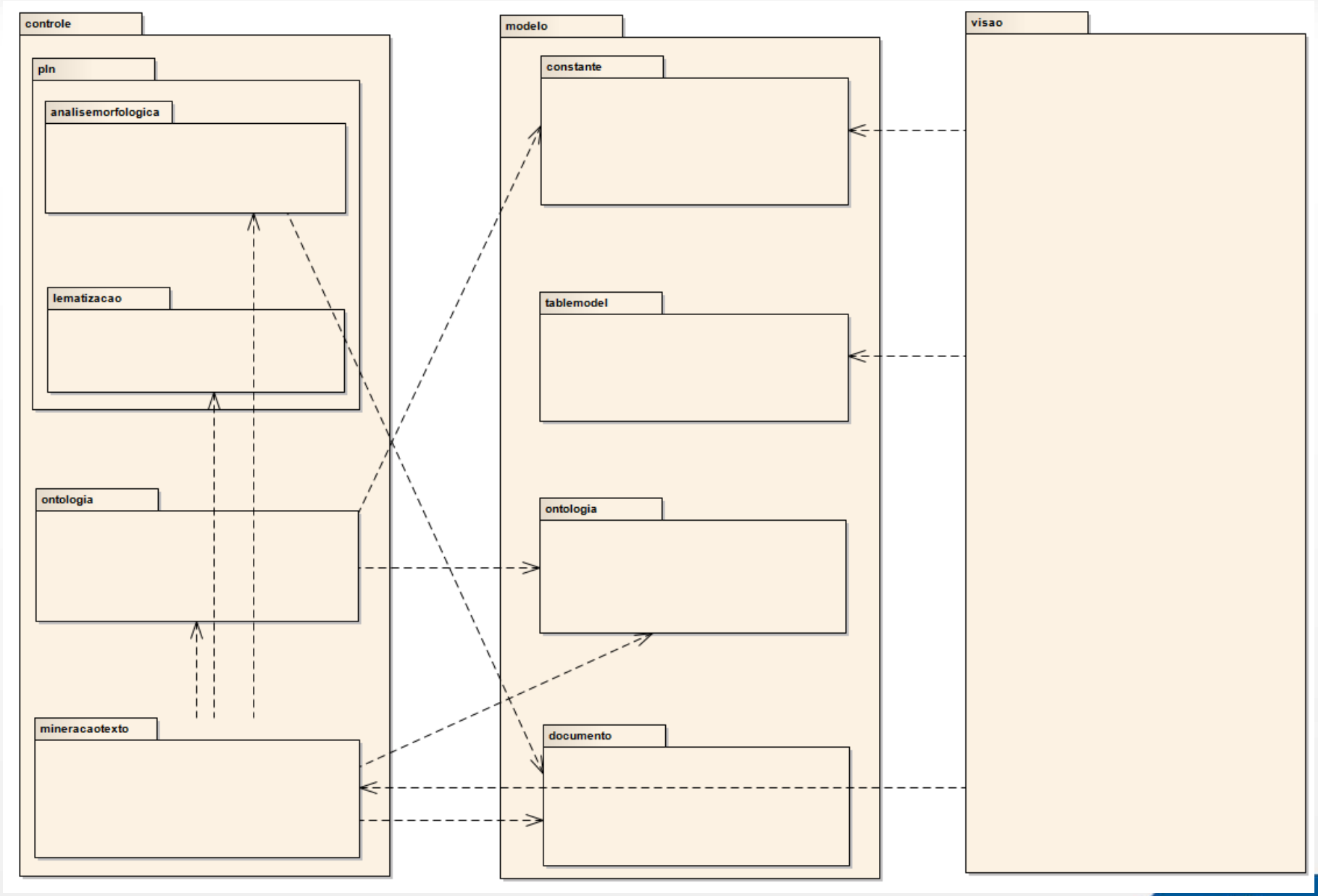
– RNF 06: A ferramenta deverá utilizar a biblioteca PTStemmer (OLIVEIRA, 2010) na versão 2.0 para lematizar as palavras contidas na base textual de trabalho atual.

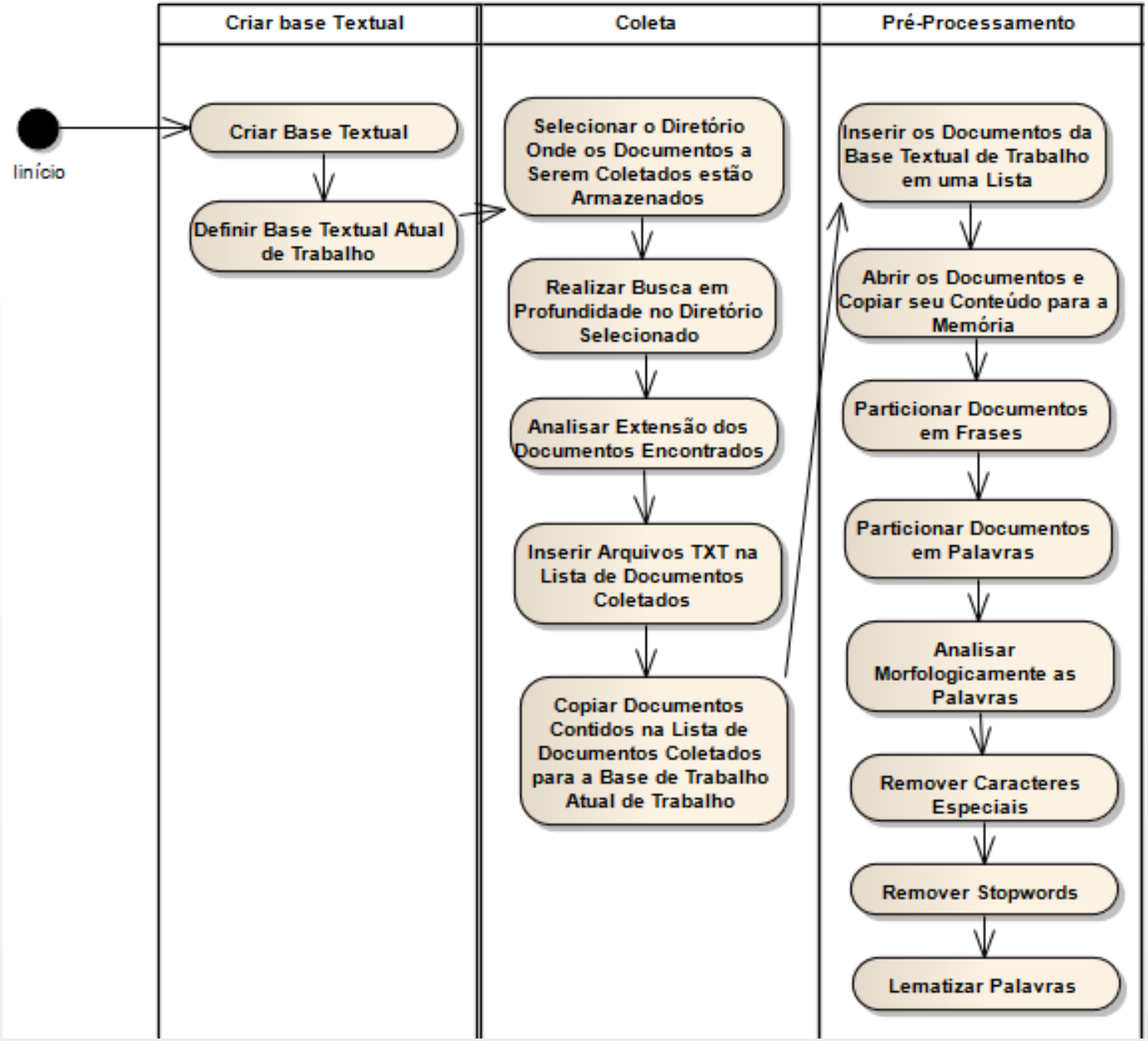
– RNF 07: A ferramenta analisará apenas documentos escritos em língua portuguesa;

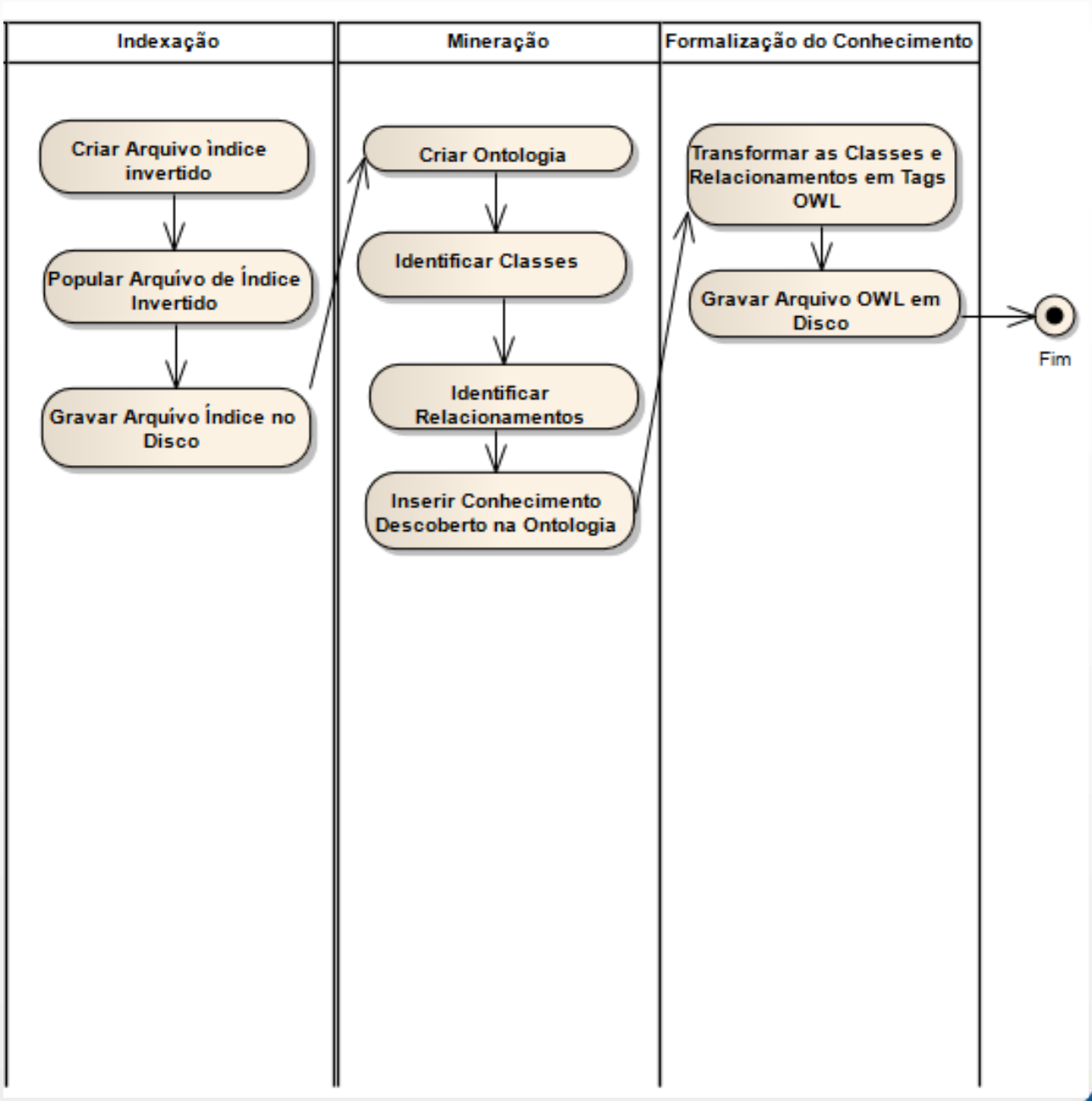
– RNF 08: A ferramenta deverá gerar uma ontologia OWL fazendo uso das *tags* definidas pelo Protégé, sendo assim possível manipulá-la através do mesmo.

Especificação


UC 03 - definir base textual
UC 04 - popular base textual
«include»
UC 05 - exibir estrutura base textual
UC 06 - exibir base textual anotada morfologicamente
UC 02 - excluir base textual
«include»
UC 01 - criar base textual
UC 07 - exibir estrutura arquivo índice
UC 08 - consultar documento por palavra
Gestor do Conhecimento
UC 09 - exibir base textual sem stopwords
UC 12 - criar base de conhecimento
«include»
UC 11 - descobrir conhecimento
UC 10 - exibir base textual lematizada


controle
pln
analisemorfologica
lematizacao
ontologia
mineracaotexto
modelo
constante
tablemodel
ontologia
documento
visao


Criar base Textual
Coleta
Pré-Processamento
início
Criar Base Textual
Definir Base Textual Atual de Trabalho
Selecionar o Diretório Onde os Documentos a Serem Coletados estão Armazenados
Realizar Busca em Profundidade no Diretório Selecionado
Analisar Extensão dos Documentos Encontrados
Inserir Arquivos TXT na Lista de Documentos Coletados
Copiar Documentos Contidos na Lista de Documentos Coletados para a Base de Trabalho Atual de Trabalho
Inserir os Documentos da Base Textual de Trabalho em uma Lista
Abrir os Documentos e Copiar seu Conteúdo para a Memória
Particionar Documentos em Frases
Particionar Documentos em Palavras
Analisar Morfologicamente as Palavras
Remover Caracteres Especiais
Remover Stopwords
Lematizar Palavras

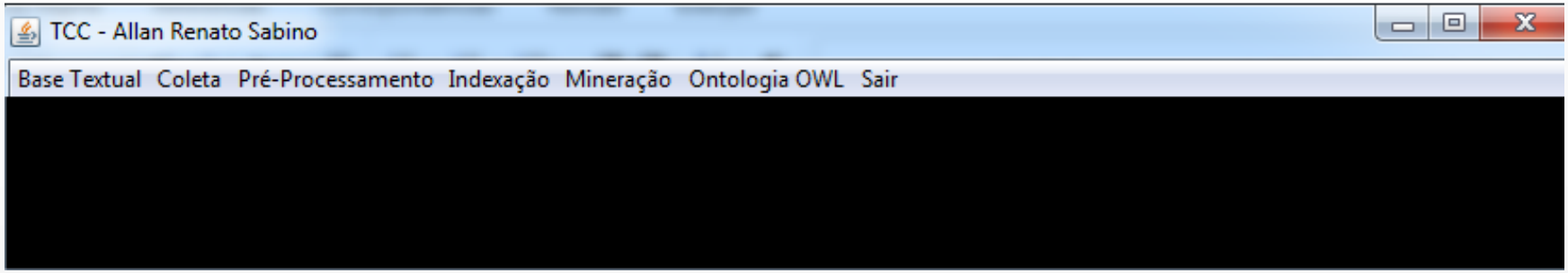

Indexação
Mineração
Formalização do Conhecimento
Criar Arquivo ìndice invertido
Popular Arquivo de Índice Invertido
Gravar Arquivo Índice no Disco
Criar Ontologia
Identificar Classes
Identificar Relacionamentos
Inserir Conhecimento Descoberto na Ontologia
Transformar as Classes e Relacionamentos em Tags OWL
Gravar Arquivo OWL em Disco
Fim

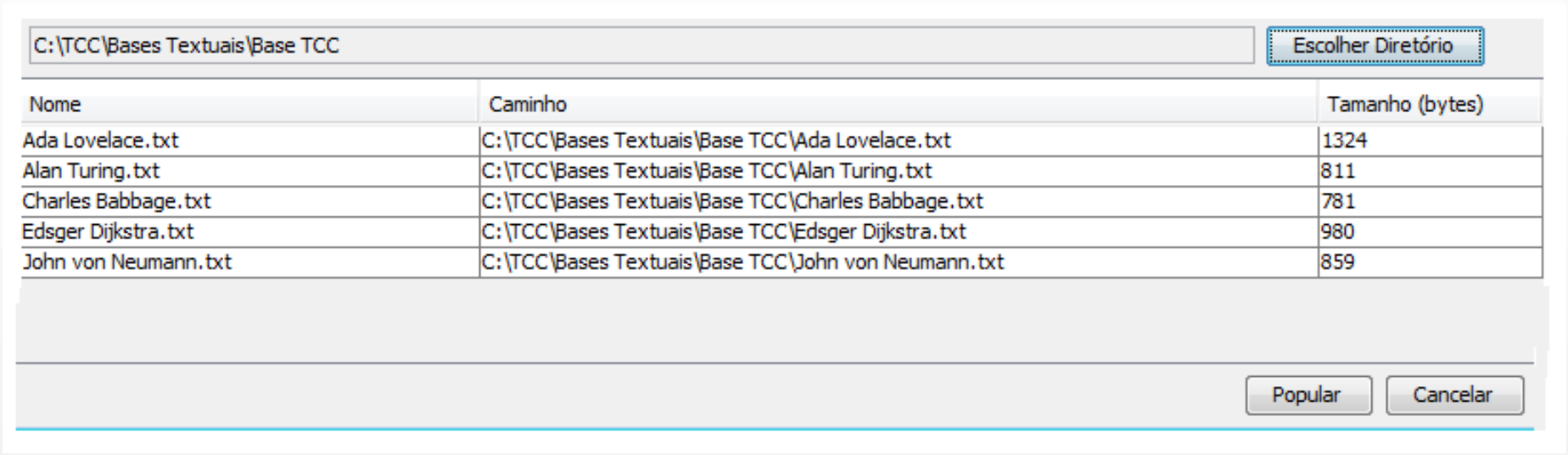
Implementação

```
1  Stemmer lematizador = new OrengoStemmer();
2  System.out.println(lematizador.getWordStem("Utilizando o algoritmo Orengo"));
3
4  Stemmer lematizador = new PorterStemmer();
5  System.out.println(lematizador.getWordStem("Utilizando o algoritmo Porter"));
6
7  Stemmer lematizador = new SavoyStemmer();
8  System.out.println(lematizador.getWordStem("Utilizando o algoritmo Savoy"));
```

```
1  ComponentFactory factory = ComponentFactory.create(new Locale("pt", "BR"));
2  Analyser  cogroo = factory.createPipe();
3
4  Document document = new DocumentImpl();
5  document.setText(documentText);
6  cogroo.analyze(document);
7
8  for (Sentence sentence : document.getSentences()) {
9    for (Token token : sentence.getTokens()) {
10
11       System.out.println(token.getLexeme());
12       System.out.println(Arrays.toString(token.getLemmas()));
13       System.out.println(token.getPOSTag());
14       System.out.println(token.getFeatures());
15   }
16 }
```

Operacionalidade da Implementação


TCC - Allan Renato Sabino
Base Textual Coleta Pré-Processamento Indexação Mineração Ontologia OWL Sair


C:\TCC\Bases Textuais\Base TCC
Escolher Diretório
Nome
Caminho
Tamanho (bytes)
Ada Lovelace.txt
C:\TCC\Bases Textuais\Base TCC\Ada Lovelace.txt
1324
Alan Turing.txt
C:\TCC\Bases Textuais\Base TCC\Alan Turing.txt
811
Charles Babbage.txt
C:\TCC\Bases Textuais\Base TCC\Charles Babbage.txt
781
Edsger Dijkstra.txt
C:\TCC\Bases Textuais\Base TCC\Edsger Dijkstra.txt
980
John von Neumann.txt
C:\TCC\Bases Textuais\Base TCC\John von Neumann.txt
859
Popular
Cancelar

Lexema	Lemas	Classe Gramatical	Inflexão
Ada_Augusta_Byron_King		Nome Própio	Feminino, Singular
Condessa_de_Lovelace		Nome Própio	Feminino, Singular
atualmente	atual	Advérbio	-
conhecida	conhecer	Verbo Particípio	Feminino, Singular
como	como	Advérbio	-
Ada_Lovelace		Nome Própio	Feminino, Singular
foi	ir	Verbo Finito	Perfeito Simples, Terceira Pessoa Singular, Indicativo
uma	um	Artigo	Feminino, Singular
matemática	matemática	Substantivo	Feminino, Singular
e	e	Conjução Coordenativa	-
escritora	escritor	Substantivo	Feminino, Singular
inglesa	inglês	Adjetivo	Feminino, Singular
e	e	Conjução Coordenativa	-
hoje	hoje	Advérbio	-
é	ser	Verbo Finito	Presente, Terceira Pessoa Singular, Indicativo
principalmente	principal	Advérbio	-
reconhecida	reconhecer	Verbo Particípio	Feminino, Singular
por	por	Preposição	-
ter	ter	Verbo Infinitivo	-
escrito	escrever	Verbo Particípio	Masculino, Singular
o	o	Artigo	Masculino, Singular
primeiro	primeiro	Adjetivo	Masculino, Singular
algoritmo	algoritmo	Substantivo	Masculino, Singular
para	para	Preposição	-
ser	ser	Verbo Infinitivo	-
processado	processar	Verbo Particípio	Masculino, Singular
por	por	Preposição	-
uma	um	Artigo	Feminino, Singular
máquina	máquina	Substantivo	Feminino, Singular
a	o	Artigo	Feminino, Singular
máquina	máquina	Substantivo	Feminino, Singular
analítica	analítico	Adjetivo	Feminino, Singular

Analisar Base Textual | Sair

```
<Ontology>
        <Declaration>
                <Class IRI="#Ada_Augusta_Byron_King"/>
        </Declaration>
        <Declaration>
                <Class IRI="#Ada_Lovelace"/>
        </Declaration>
        <Declaration>
                <Class IRI="#Alan_Mathison_Turing"/>
        </Declaration>
        <Declaration>
                <ObjectProperty IRI="#foi"/>
        </Declaration>
                <ObjectPropertyDomain>
                        <ObjectProperty IRI="#foi"/>
                        <Class IRI="#Alan_Mathison_Turing"/>
                </ObjectPropertyDomain>
                <ObjectPropertyRange>
                        <ObjectProperty IRI="#foi"/>
                        <Class IRI="#computação"/>
                </ObjectPropertyRange>
        <Declaration>
                <Class IRI="#Albert_Einstein"/>
        </Declaration>
        <Declaration>
                <Class IRI="#Anotações"/>
        </Declaration>
        <Declaration>
                <Class IRI="#Babbage"/>
        </Declaration>
```

Exibir Ontologia OWL | Sair

Resultados e Discussões

- Estudo de caso:
  - Cinco arquivos;
  - Seiscentas e noventa e sete palavras.

- Validação com a ferramenta Protégé.

- Classes identificadas corretamente;
- Limitações ao identificar relacionamentos.

Características	Trabalho Desenvolvido	Baségio (2007)	Faria e Girardi (2010)	Silva (2004)
Identificar estruturas ontológicas	X	X	-	-
Popular ontologia OWL	-	-	X	-
Criar Ontologia OWL	X	X	-	-
AM	-	-	-	X
Remover *stopwords*	X	X	-	X
Lematizar palavras	X	X	-	X
Realizar mineração	X	X	X	X
Realizar análise morfológica	X	-	-	-
Realizar coleta	X	-	-	-
Criar arquivo de índice	X	X	-	-

Conclusões

- Todos os requisitos propostos foram atendidos;
- Duas decisões tomaram influenciaram a qualidade do conhecimento descoberto;
- Ferramentas utilizadas mostraram-se de vital importância.

Extensões

- Permitir o cadastro de *stopwords*;
- Permitir o cadastro de caracteres especiais;
- Permitir a escolha do algoritmo de lematização utilizado.

- Utilizar o modelo de representação baseado em termos;
- Permitir a identificação de abreviações;
- Permitir a identificação de símbolos da internet.

- Utilizar sinônimos durante o pré-processamento para diminuir a dimensionalidade dos documentos;
- Utilizar análise sintática para a descoberta de conhecimento;
- Implementar um analisador morfológico próprio.

- Identificar herança de classes;
- Popular ontologia com objetos a partir do conhecimento da base textual;
- Identificar propriedades das propriedades (transitiva, simétrica, entre outras).

- Identificar restrições nas propriedades;
- Identificar equivalências entre classes, ou seja, classes com nomes diferentes, mas mesma semântica;
- Identificar equivalência de objetos.

Demonstração