

Descrição do projeto BibSys

O *BibSys* é um programa que tem por objetivo realizar o gerenciamento de arquivos *bibtex* (.bib). Esse tipo de arquivo armazena a descrição bibliográfica de um artigo científico, livro, site, entre outros. Existem várias entradas para caracterizar as referências, sendo todas precedidas pelo caractere (@)

Exemplos:

<pre>@article{, author = {}, title = {}, journal = {}, volume = {}, number = {}, pages = {}, year = }</pre>	<pre>@book{, author = {}, title = {}, series = {}, volume = {}, publisher = {}, address = {}, year = {}, }</pre>	<pre>@misc{, author = {}, title = {}, text = {}, year = {}, url = {}, }</pre>
---	--	---

Veja agora um exemplo instanciado para a entrada “@ARTICLE”:

```
@ARTICLE{Carnielli:1999,
  author = {W.A. Carnielli and J. Marcos},
  title = {Limits for paraconsistent calculi},
  journal = {Notre Dame Journal of Formal Logic},
  year = {1999},
  volume = {40},
  number = {3},
  pages = {375--390},
}
```

Um arquivo “.bib” por ter uma ou mais referências, as quais são identificadas pela *tag* bibkey. No exemplo acima, o valor (Carnielli:1999) identifica a referência bibliográfica. Já as demais *tags* descrevem a referência de acordo com o tipo. Nota-se também que esse exemplo utiliza ({} ,) para separar as *tags* no arquivo. Em cada base o formato dos arquivos pode ser diferente. Por exemplo, ao invés de utilizar {} para atribuir um valor a uma *tag*, pode ser utilizado ". As bases de pesquisa mais consultadas são:

- IEEE Xplore www.ieeexplore.ieee.org
- ScienceDirect www.sciencedirect.com
- Scopus www.scopus.com
- Springer www.springerlink.com
- Web of Science www.isiknowledge.com

Para permitir o gerenciamento desse tipo de arquivo, o programa *BibSys* deve implementar as seguintes funcionalidades:

- 1) **Padronização de arquivos.** Para isso, uma funcionalidade deve ser implementada para padronizar (formatar) os arquivos (.bib) conforme o exemplo abaixo:

```
@ARTICLE{Carnielli:1999,  
  author      = {W.A. Carnielli and J. Marcos},  
  title       = {Limits for paraconsistent calculi},  
  journal     = {Notre Dame Journal of Formal Logic},  
  year        = {1999},  
  volume      = {40},  
  number      = {3},  
  pages       = {375-390},  
}
```

Requisitos para padronização:

1. Todas as referências do arquivo deve ter *tags* internas em linhas diferentes;
 2. Todos os valores das *tags* devem ser envolvidos por {}, ou seja, o programa deve substituir as "" por {}, quando necessário;
 3. Espaçamento:
 1. Antes das *tags* devem existir dois espaços em branco;
 2. Até o sinal de (=) deve existir 16 caracteres. Preencher as *tags* com espaço em branco;
 3. Entre o sinal de (=) e a ({}) deve existir apenas um espaço em branco;
 4. O valor da *tag* deve estar entre chaves e sem espaços em branco no início e no fim. Exemplo: {valor dessa *tag*}.
- 2) **Comparação de arquivos.** Para isso, o programa deve permitir abrir dois arquivos “.bib” e permitir apontar quais referências estão em um arquivo A que não estão em B e vice-versa. Como resultado, o programa pode indicar nos arquivos (destacar em cores) ou apresentar os resultados em outros arquivos (saída externa).
 - 3) **Concatenação de arquivos.** Para isso, o programa deve permitir abrir dois arquivos “.bib” e permitir juntar as referências de dois arquivos A e B. Como resultado, um terceiro arquivo deve ser gerado.
 - 4) **Geração automática de *bibkey*.** Para isso, o programa levará em conta os seguintes critérios:

- a. A chave será formada pelo no dos autores + “:” + ano. Quando um autor (ou grupo) tiver mais do que uma referência naquele ano, o programa deverá diferenciá-las adicionando letras ao final da chave.
 - b. Regras de formação:
 - i. Apenas um autor:
 1. Fulano da Silva → *bibkey*: silva:2017;
 - ii. Dois autores:
 1. Fulano da Silva e Ciclano da Rocha → *bibkey*: silva.rocha:2017;
 - iii. Três ou mais autores:
 1. Fulano da Silva e Ciclano da Rocha e Beltrano da Costa → *bibkey*: silva.etal:2017;
 - iv. Havendo duplicatas de *bibkey*, o programa deve adicionar letras para diferenciá-las.
 1. silva:2017;
 2. silva:2017a;
 3. silva:2017b.
- 5) **Ordenação de referências pela chave (*bibkey*).** Para isso, o programa deve permitir abrir um arquivo e ordenar esse arquivo. Como resultado, o programa deve informar se o arquivo será ordenado ou se a ordenação será apresentada em outro arquivo.
 - 6) **Operações elementares.** O programa deve permitir abrir, salvar e fechar um arquivo “.bib”. Outras operações podem ser implementadas.
 - 7) **Casos omissos.** Os casos omissos serão tratados por cada grupo.

Requisitos para a implementação em Java:

1. Ao chamar o programa, uma tela de *splash* deve ser exibida e permanecer ativa por 5 segundos. Essa tela deve apresentar o nome do sistema (*BibSys*), dos alunos em ordem alfabética e a versão do sistema (2017);
2. Em seguida, o sistema é minimizado na barra de tarefa ao lado do relógio (*SystemTray*);
3. Quando o usuário clicar no ícone, um menu é exibido e a tela do *BibSys* é exibida.
4. A confecção da tela é livre.

Referências

- <http://docs.oracle.com/javase/tutorial/uiswing/TOC.html>
- <http://www.bibtex.org/>