

Universidade do Minho

Escola de Engenharia  
Departamento de Informática

Processamento de Linguagens

# PLIKIPÉDIA

Pedro Faria, A60998  
Mariana Medeiros, A61041  
Miguel Pinto, A61049

9 de Abril de 2014

# Conteúdo

<b>1</b>	<b>Introdução</b>	<b>4</b>
1.1	Descrição do Problema e Implementação . . . . .	4
<b>2</b>	<b>Processamento da Wikipédia</b>	<b>5</b>
2.1	Expressões Regulares . . . . .	5
2.2	Estados da Aplicação . . . . .	5
2.3	Módulos da Aplicação . . . . .	8
2.4	Estruturas de Dados . . . . .	8
<b>3</b>	<b>HTML</b>	<b>10</b>
3.1	Página Inicial . . . . .	10
3.2	Artigo . . . . .	11
3.2.1	Informações Relevantes . . . . .	11
3.2.2	Secções . . . . .	11
3.2.3	Links Internos . . . . .	12
3.2.4	Links Externos . . . . .	12
3.3	Relatório . . . . .	13
3.4	Grupo . . . . .	14
<b>4</b>	<b>Conclusões</b>	<b>15</b>

***Resumo***

---

O presente relatório descreve todo o processo de desenvolvimento, tomadas de decisão e resultados obtidos na resolução do primeiro projeto prático proposto no âmbito da Unidade Curricular de Processamento de Linguagens. O enunciado aqui resolvido aborda o tema referente ao processamento de informação extraída da Wikipédia - variante 1. O produto final obtido consiste num processador de ficheiros .xml, descarregados a partir da base de conhecimento da Wikipédia, cujo output são ficheiros .html que apresentam os dados obtidos de uma forma agradável e bem estruturada.

**Palavras chave:** Wikipédia, XML, HTML, FLEX, Expressões Regulares, Estruturas de Dados.

---

# 1 Introdução

De forma a colocar em prática os conhecimentos adquiridos até ao momento nas aulas teóricas e teórico-práticas da já referida Unidade Curricular, fizemos uso da ferramenta Flex - The Fast Lexical Analyzer e tentamos definir da melhor forma possível as expressões regulares utilizadas, no sentido de obter um bom resultado no processamento da informação.

Neste relatório serão apresentadas e detalhadamente justificadas todas as tomadas de decisão relativas à realização deste projeto, assim como uma sucinta explicação das funcionalidades presentes nas páginas HTML geradas.

## 1.1 Descrição do Problema e Implementação

A decisão referente a qual dos enunciados de trabalho escolher acabou por recair na do processamento da Wikipédia - variante 1.

A Wikipédia é uma gigantesca base de conhecimento online disponível nas mais variadas línguas, o que torna o seu processamento um desafio muito interessante.

Neste trabalho pretende-se que criemos um executável que torne possível processar páginas da Wikipédia, ou seja, é pedido que se desenvolva, em Flex, um filtro para estruturar, num site HTML, um conjunto de informação extraída da Wikipédia.

Deste modo, foi necessário exportar as páginas da Wikipédia pretendidas, num formato passível de transformação (XML). Para que isto fosse possível, utilizamos o Special Export, tal como está especificado no enunciado do projeto.

Iniciamos o projeto com a definição das várias expressões regulares que serviram para fazer o parsing do ficheiro .xml que contém as informações descarregadas da Wikipédia, sendo que estas terão de conseguir filtrar de forma simples e clara toda a informação necessária como, por exemplo, o título, o autor da última revisão, a data da última revisão entre outros elementos que compeltam a resolução deste trabalho.

Após termos todas as expressões regulares construídas e o estado do Flex definido, passamos para o desenvolvimento da estrutura da página HTML, onde de uma forma simples e organizada podemos observar toda a informação filtrada da Wikipédia.

## 2 Processamento da Wikipédia

Para processar o arquivo .xml, e com o objetivo de se obter os resultados esperados, recorreu-se ao analisador léxico *FLEX*. Nos pontos seguintes iremos abordar a forma como foram definidas as expressões regulares, os estados do FLEX e a estrutura de dados usada para guardar alguns valores.

### 2.1 Expressões Regulares

As expressões regulares definidas tratam um ficheiro .xml com o objetivo de obter as informações necessárias para a construção de uma página *HTML*. As expressões regulares que definimos estão presentes no ficheiro .fl e seguem a seguir.

Nesta tarefa, recorreremos ao tutorial da Wikipédia que nos ajudou a perceber alguns dos prefixos do xml.

**Título** <title>.\*</title>

**Data última revisão** <timestamp>.\*</timestamp>

**Autor última revisão** <username>.\*</username>

**Secção** ==.\*==

**SubSecção** ===.\*===

**SubSubSecção** ====.\*====

**SubSubSubSecção** =====.\*=====

**Link Interno** [[.\*| ou [[.\*]]

**Link Externo** http://.\*

### 2.2 Estados da Aplicação

Para o desenvolvimento do analisador léxico recorreremos a um autómato constituído por vários estados: *PAGE*, *TITLE*, *REVISION*, *DATE*, *AUTHOR*, *LINKINT*, *LINKEXT*, *SECTION1*, *SECTION2*, *SECTION3* e *SECTION4*.

#### Estado INITIAL

Neste estado procura-se pela tag <page> e assim que ela é encontrada entramos no estado *PAGE* e é também inicializada a estrutura de dados *Página pag*, que irá ser abordada mais adiante neste relatório.

```
"<page>" {
    BEGIN PAGE; pag = inicializaPagina();
}
```

### Estado PAGE

Neste estado encontram-se algumas inicializações de outros estados (TITLE, REVISION e INITIAL) depois de se verificar se cada uma das *tags* é coincidente com as definidas no ficheiro FLEX. A expressão `.|[\n\t]` seleciona a parte textual que "não tem interesse" para a aplicação realizada.

Quando é encontrada a *tag* `</page>` é criada uma página *HTML* para a página analisada: é inserido o título da página numa lista ligada para no final ser criado um índice de todas as páginas processadas. No final é chamado o estado INITIAL para verificar, caso exista, uma nova página.

```
<PAGE>{
    "<title>"          BEGIN TITLE;
    "<revision>"        BEGIN REVISION;
    ".|[\n\t]"         ;
    "</page>"           criaFicheiroHTML(pag); insereTituloIndice(lpags,pag->titulo);
BEGIN INITIAL;
}
```

### Estado TITLE

Aqui encontramos o título da página que estamos a analisar e guardamo-lo na nossa estrutura.

De modo a evitar captar páginas que contém como título *Categoria:Astronomia*, por exemplo, e que são irrelevantes pois apenas listam vários temas relacionados com Astronomia, usamos a expressão `.*:.*` que se for coincidente descarta essa página, ou seja, vai verificar uma nova página (chamada do estado INITIAL).

```
<TITLE>{
    .*:.*              BEGIN INITIAL;
    "[^<]*"            insereTitulo(pag,yytext);
    "</title>"          BEGIN PAGE;
}
```

### Estado REVISION

Este estado contém grande parte da informação que queremos filtrar e prende-se com a última revisão efetuada à página em questão. É o corpo do artigo.

Sempre que é encontrado alguma expressão igual às definidas abaixo, é inicializado um estado, retirada a informação que pretendemos, caso exista, ou então volta ao estado anterior - PAGE. É ainda aqui que se tenta encontrar os links internos da página, sendo este depois inserido na lista de links internos.

```
<REVISION>{
    ".|[\n\t]"        ;
    "<timestamp>"      BEGIN DATE;
    "<username>"       BEGIN AUTHOR;
    "=="              BEGIN SECTION1;
    "==="             BEGIN SECTION2;
    "===="            BEGIN SECTION3;
    "====="           BEGIN SECTION4;
```

```

[[[^\:|]]+[/]      insereLinkInt(pag,yytext+2);
[[[^\:|]]+[/|]]     insereLinkInt(pag,yytext+2);
"http://"          BEGIN LINKEXT;
"</revision>"      BEGIN PAGE;
}

```

### Estado DATE

É obtida a data da última edição, que é guardada na estrutura de dados. No final volta-se ao estado REVISION.

```

<DATE>{
    [^\<]*          insereData(pag,yytext);
    «/timestamp>"   BEGIN REVISION;
}

```

### Estado AUTHOR

Tal como com a data da última revisão, o valor obtido pela expressão é guardado e depois volta ao estado REVISION.

```

<AUTHOR>{
    [^\<]*          insereAutor(pag,yytext);
    «/username>"    BEGIN REVISION;
}

```

### Estados SECTION1, SECTION2, SECTION3 e SECTION4

Nestes estados são obtidos os nomes das secções existentes em cada página da Wikipédia. Como os estados são bastante parecidos decidimos agrupá-los. Existe ainda uma expressão que deteta quando uma secção está errada, voltando assim ao estado REVISION.

Cada secção originada é inserida numa lista de secções onde é identificado o tipo de secção. Finalmente, detetada a expressão regular final, volta ao estado REVISION.

```

<SECTION1>{
    .|[\n\t]        BEGIN REVISION;
    [^\=\\n]*       insereSeccao(0,pag,yytext);
    ==              BEGIN REVISION;
}

```

```

<SECTION2>{
    [^\=&]+          insereSeccao(1,pag,yytext);
    .|[\n\t]        ;
    ===              BEGIN REVISION;
}

```

```

<SECTION3>{
    [^\=]*           insereSeccao(2,pag,yytext);
    ====            BEGIN REVISION;
}

```

```

}

<SECTION4>{
    [^=]*      insereSeccao(3,pag,yytext);
    =====  BEGIN REVISION;
}

```

### Estado LINKEXT

Aqui é obtido o URL total do link externo até encontrar um espaço. Depois de inserido na lista de links externos da página, o estado REVISION é chamado.

```

<LINKEXT>{
    [a-zA-Z0-9/._-?=&;+-%]+      insereLinkExt(pag,yytext); BEGIN REVISION;
}

```

## 2.3 Módulos da Aplicação

A aplicação *PLIKIPÉDIA* desenvolvida tem por base os seguintes módulos:

**parserXML.fl** Encontra-se o código fonte para fazer a análise léxica aos ficheiros XML.

**linkedlist.h** Contém o código fonte das listas ligadas genéricas e suas funções.

**auxstruct.h** Tem o código para inserir dados na estrutura de dados Pagina.

**htmlpage.h** É onde se encontra o código que gera as páginas .html de cada página da Wikipédia, além do índice de títulos lidos.

**makefile** Ferramenta com a configuração de compilação dos ficheiros acima descritos.

## 2.4 Estruturas de Dados

Como verificamos que era necessário guardar alguns dados, recorremos a módulos de listas ligadas genéricas. Assim que a aplicação é executada, é inicializada uma lista ligada que irá salvar os títulos de todas as páginas válidas, para que no fim da leitura dos ficheiros .xml seja criada uma página que contém o título de todos os artigos lidos.

Durante a análise léxica de uma página, recorreu-se a uma estrutura de dados denominada Pagina onde foram guardadas várias informações relevantes. Contém também três listas ligadas genéricas que visam guardar, respetivamente, as secções, subsecções, etc. e também os links internos e externos.

```

typedef struct sPagina {
    char* titulo;
    char* data;
    char* autor;

```



```
    LinkedList seccoes;  
    LinkedList linkint;  
    LinkedList linkext;  
} *Pagina, NPagina;
```

Depois de criada a página HTML referente à última página abordada a estrutura de dados é limpa e inicializada para recolher os novos dados de outra página, caso estes existam.

### 3 HTML

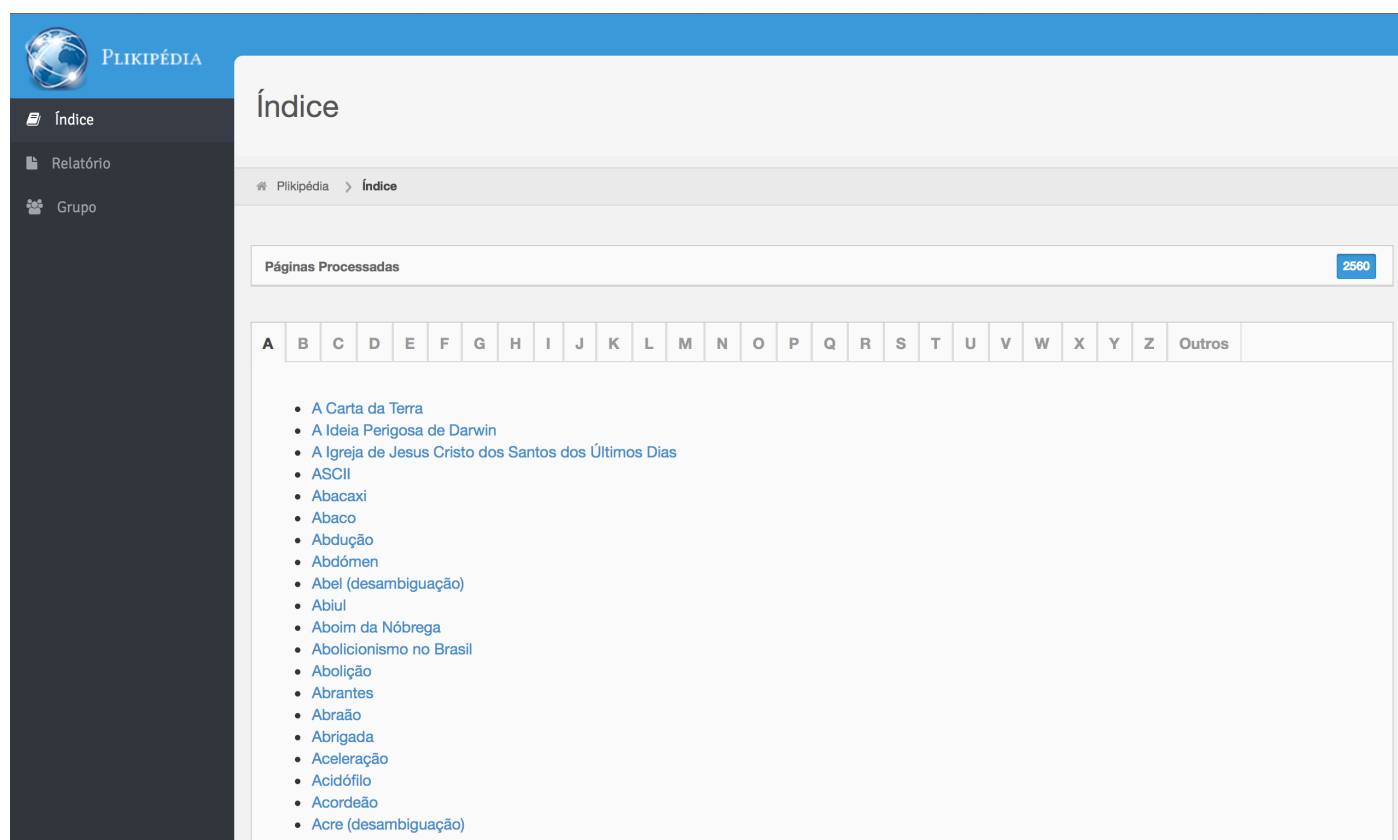
De forma a tornar o nosso projeto mais completo, o nosso grupo decidiu acatar a possibilidade sugerida no enunciado e geramos uma página HTML para cada uma das páginas existentes no ficheiro XML descarregado.

A escolha do layout HTML incidiu, sobretudo, na funcionalidade e simplicidade do mesmo. Deste modo, optamos por recorrer a um tema fornecido pela framework Bootstrap.

#### 3.1 Página Inicial

A Página Inicial é constituída por um índice de títulos das páginas exportadas da Wikipédia. Este índice é apresentado alfabeticamente ordenado e dividido em panéis, cada um deles referente a uma letra do alfabeto, de maneira a tornar mais acessível a procura. É também assinalado o número total de páginas de Wikipédia processadas.

A partir da Página Inicial e de todas as outras, aliás, é possível, a qualquer momento, aceder a outros elementos informativos (relatório do projeto e os elementos do grupo que o realizaram), graças a uma barra de navegação lateral.



## 3.2 Artigo

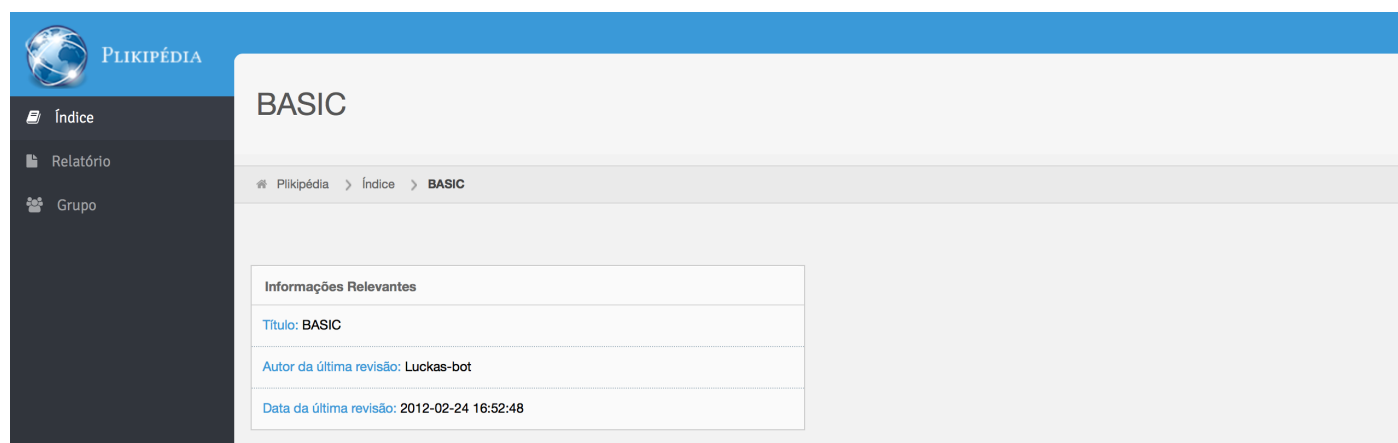
Ao seleccionar um título na Página Inicial, é mostrada a informação com ele relacionada, numa outra página, individualmente apresentada.

Cada uma destas páginas está dividida em quatro blocos: Informações Relevantes, Secções, Links Internos e Links Externos.

Ao clicar no título da página, o utilizador é encaminhado para a página original da Wikipédia, com o respetivo conteúdo original.

### 3.2.1 Informações Relevantes

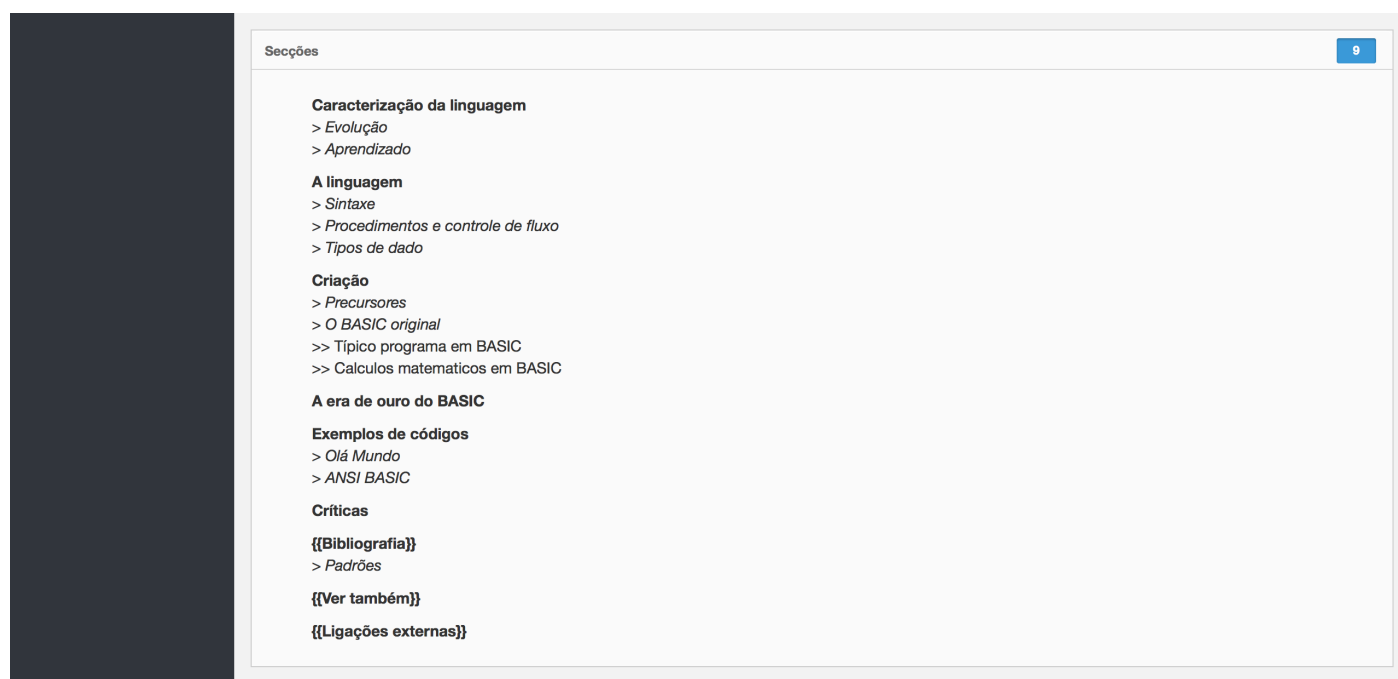
O bloco correspondente às Informações Relevantes contém informação relativa ao título do artigo, ao autor da sua última revisão e à data em que essa revisão foi realizada.



Informações Relevantes
Título: BASIC
Autor da última revisão: Luckas-bot
Data da última revisão: 2012-02-24 16:52:48

### 3.2.2 Secções

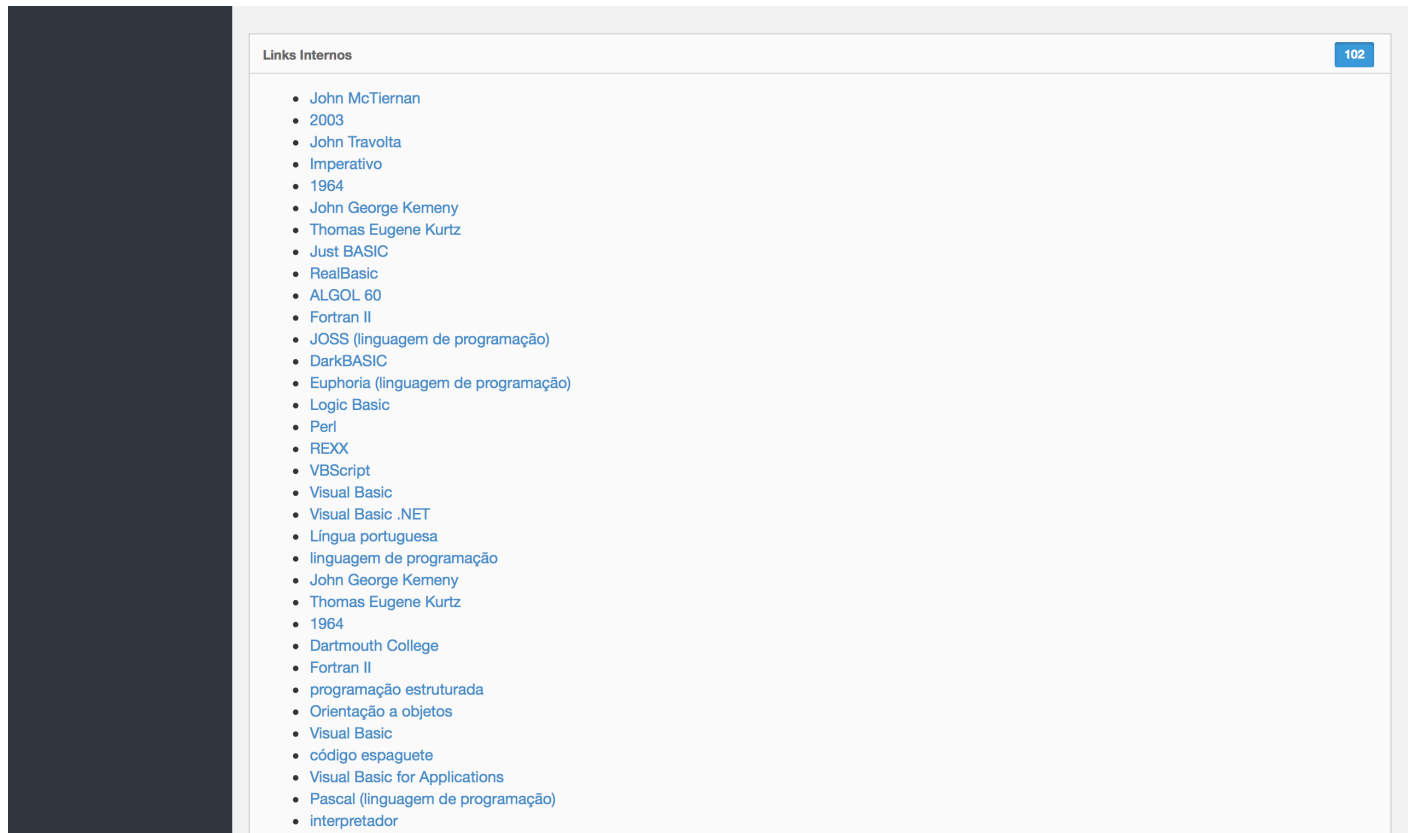
O bloco das Secções indica o número de secções e explicita quais elas são.



Secções
<b>Caracterização da linguagem</b> <ul style="list-style-type: none"><li>&gt; <i>Evolução</i></li><li>&gt; <i>Aprendizado</i></li></ul>
<b>A linguagem</b> <ul style="list-style-type: none"><li>&gt; <i>Sintaxe</i></li><li>&gt; <i>Procedimentos e controle de fluxo</i></li><li>&gt; <i>Tipos de dado</i></li></ul>
<b>Criação</b> <ul style="list-style-type: none"><li>&gt; <i>Precursores</i></li><li>&gt; <i>O BASIC original</i></li><li>&gt;&gt; <i>Típico programa em BASIC</i></li><li>&gt;&gt; <i>Calculos matematicos em BASIC</i></li></ul>
<b>A era de ouro do BASIC</b>
<b>Exemplos de códigos</b> <ul style="list-style-type: none"><li>&gt; <i>Olá Mundo</i></li><li>&gt; <i>ANSI BASIC</i></li></ul>
<b>Críticas</b>
<b>{{Bibliografia}}</b> <ul style="list-style-type: none"><li>&gt; <i>Padrões</i></li></ul>
<b>{{Ver também}}</b>
<b>{{Ligações externas}}</b>

### 3.2.3 Links Internos

O bloco dos Links Internos refere o número de links internos encontrados no corpo da página processada e lista-os. Estes links levam o utilizador para outras páginas mas sempre no domínio da Wikipédia, daí serem denominados por links internos.

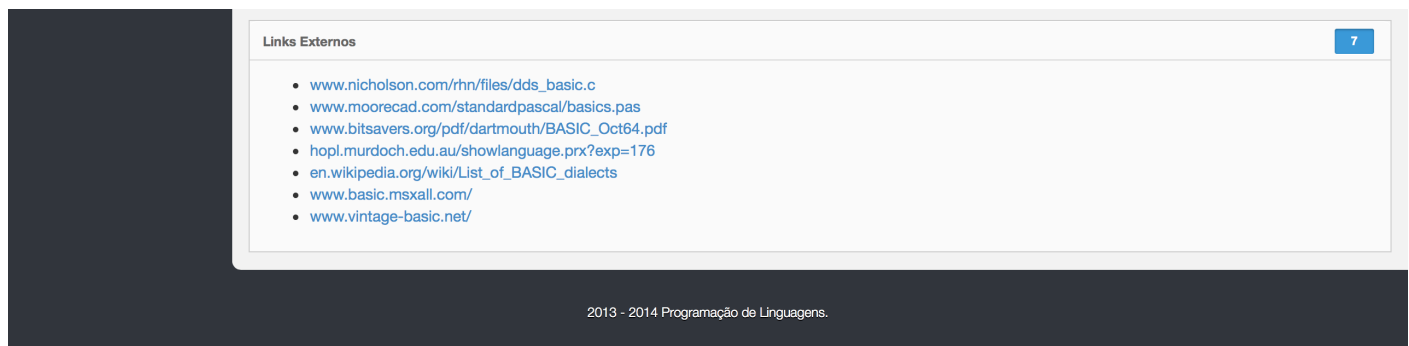


Links Internos 102

- [John McTiernan](#)
- [2003](#)
- [John Travolta](#)
- [Imperativo](#)
- [1964](#)
- [John George Kemeny](#)
- [Thomas Eugene Kurtz](#)
- [Just BASIC](#)
- [RealBasic](#)
- [ALGOL 60](#)
- [Fortran II](#)
- [JOSS \(linguagem de programação\)](#)
- [DarkBASIC](#)
- [Euphoria \(linguagem de programação\)](#)
- [Logic Basic](#)
- [Perl](#)
- [REXX](#)
- [VBScript](#)
- [Visual Basic](#)
- [Visual Basic .NET](#)
- [Língua portuguesa](#)
- [linguagem de programação](#)
- [John George Kemeny](#)
- [Thomas Eugene Kurtz](#)
- [1964](#)
- [Dartmouth College](#)
- [Fortran II](#)
- [programação estruturada](#)
- [Orientação a objetos](#)
- [Visual Basic](#)
- [código espaguete](#)
- [Visual Basic for Applications](#)
- [Pascal \(linguagem de programação\)](#)
- [interpretador](#)

### 3.2.4 Links Externos

O bloco dos Links Externos em tudo se assemelha ao bloco dos Links Internos, à exceção que estes links remetem o utilizador para páginas fora da base de conhecimento da Wikipédia.



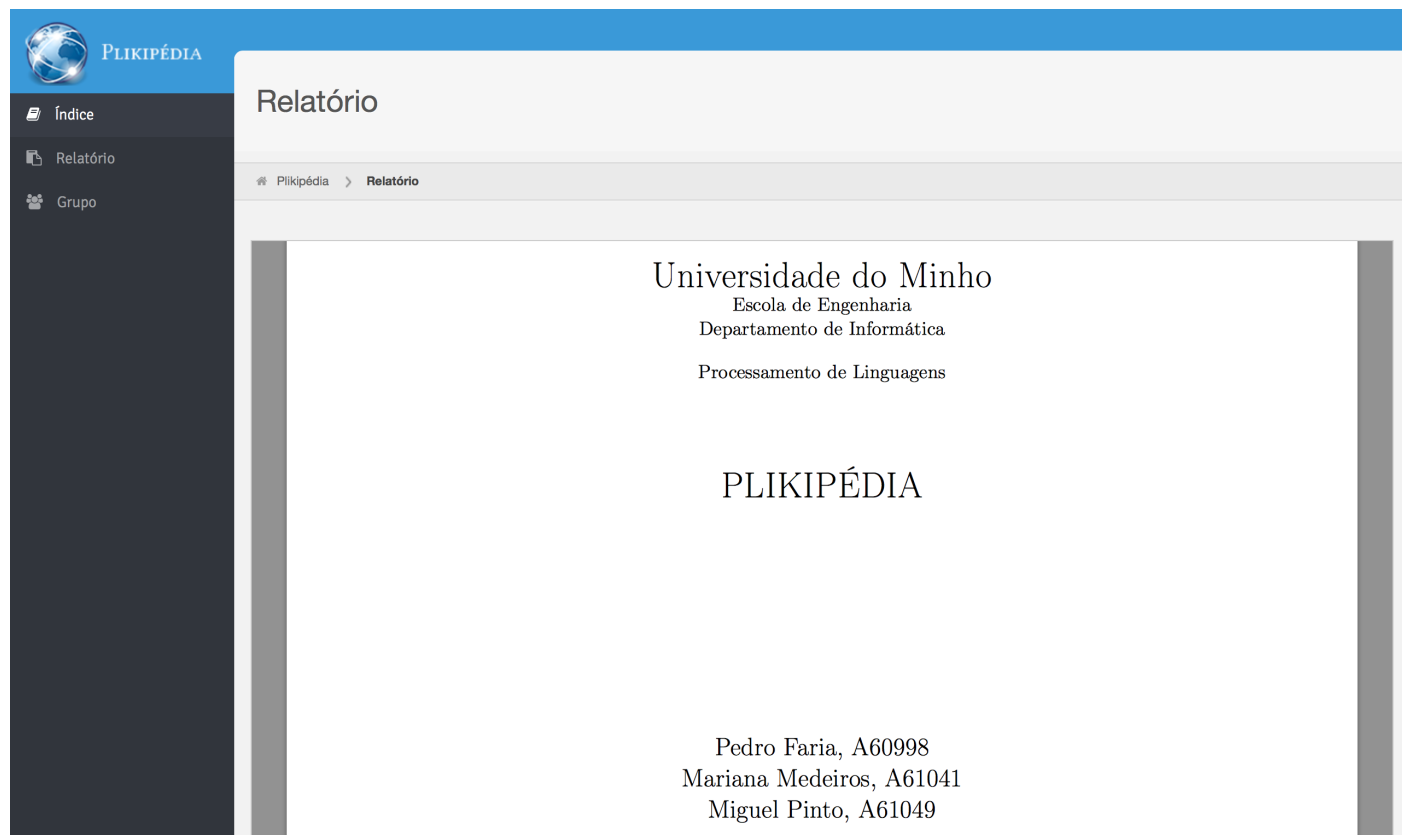
Links Externos 7

- [www.nicholson.com/rhn/files/dds\\_basic.c](http://www.nicholson.com/rhn/files/dds_basic.c)
- [www.moorecad.com/standardpascal/basics.pas](http://www.moorecad.com/standardpascal/basics.pas)
- [www.bitsavers.org/pdf/dartmouth/BASIC\\_Oct64.pdf](http://www.bitsavers.org/pdf/dartmouth/BASIC_Oct64.pdf)
- [hopt.murdoch.edu.au/showlanguage.prx?exp=176](http://hopt.murdoch.edu.au/showlanguage.prx?exp=176)
- [en.wikipedia.org/wiki/List\\_of\\_BASIC\\_dialects](http://en.wikipedia.org/wiki/List_of_BASIC_dialects)
- [www.basic.msxall.com/](http://www.basic.msxall.com/)
- [www.vintage-basic.net/](http://www.vintage-basic.net/)

2013 - 2014 Programação de Linguagens.

### 3.3 Relatório

Uma das tabs presentes na barra de navegação lateral é a do presente relatório.



### 3.4 Grupo

Por fim, a última tab da barra de navegação rápida é a que apresenta os elementos do grupo que realizou este projeto prático.

The screenshot shows the Plikipédia web application interface. On the left is a dark sidebar with a blue header containing the Plikipédia logo and name. Below the header are three navigation items: 'Índice' (with a document icon), 'Relatório' (with a document icon), and 'Grupo' (with a group of people icon). The main content area has a light gray background. At the top of this area is the title 'Grupo' in a large, bold font. Below the title is a breadcrumb trail: 'Plikipédia > Grupo'. The main content area features three portrait photos of team members arranged horizontally. Below each photo is the member's name and ID number. The first member is Pedro Faria (A60998), the second is Mariana Medeiros (A61041), and the third is Miguel Pinto (A61049). At the bottom of the page, there is a dark gray footer with the text '2013 - 2014 Programação de Linguagens.'

## 4 Conclusões

Em modo de conclusão, vemos este projeto como mais um forma de pôr em prática os conhecimentos adquiridos e as ferramentas disponibilizadas na Unidade Curricular de Processamento de Linguagens. Como o enunciado deste trabalho prático implica um domínio de todos os conceitos abordados nas aulas, este ajudou a consolidar os conhecimentos e a combater algumas dificuldades que tínhamos anteriormente.

Apesar da abordagem deste enunciado ser bastante centrada no formato fixo dos ficheiros descarregados da Wikipédia, com umas pequenas alterações, poderemos reutilizar este código para outros futuros fins.

O projeto foi desenvolvido de maneira a conseguir dar resposta ao tema que nos propusemos abordar e vendo o resultado final, todos os elementos do grupo concordam que fomos bem sucedidos.