

LI3 Project - MiEI 16/17
grupo69

Generated by Doxygen 1.8.13

Contents

1	Data Structure Index	1
1.1	Data Structures	1
2	File Index	3
2.1	File List	3
3	Data Structure Documentation	5
3.1	article Struct Reference	5
3.1.1	Detailed Description	5
3.1.2	Field Documentation	5
3.1.2.1	id	5
3.1.2.2	nWords	6
3.1.2.3	revisions	6
3.1.2.4	size	6
3.1.2.5	title	6
3.2	revision Struct Reference	6
3.2.1	Detailed Description	6
3.2.2	Field Documentation	7
3.2.2.1	id	7
3.2.2.2	timestamp	7
3.3	TCD_istruct Struct Reference	7
3.3.1	Detailed Description	7
3.3.2	Field Documentation	7
3.3.2.1	allArticles	8
3.3.2.2	allRevisions	8
3.3.2.3	articles	8
3.3.2.4	uniqueArticles	8
3.3.2.5	users	8
3.4	user Struct Reference	8
3.4.1	Detailed Description	9
3.4.2	Field Documentation	9
3.4.2.1	contributions	9
3.4.2.2	id	9
3.4.2.3	username	9

4 File Documentation	11
4.1 C:/Users/Vitor Castro/Desktop/grupo69/articles.c File Reference	11
4.1.1 Function Documentation	12
4.1.1.1 get_all_articles()	12
4.1.1.2 get_all_revisions()	12
4.1.1.3 get_article_timestamp()	13
4.1.1.4 get_article_title()	13
4.1.1.5 get_unique_articles()	14
4.1.1.6 getTitlesWithPrefix()	14
4.1.1.7 getTop20LargestArticles()	15
4.1.1.8 getTopNArticlesWithMoreWords()	15
4.1.1.9 isTitlePrefix()	16
4.1.1.10 onPageArticles()	16
4.1.1.11 wordCounter()	17
4.2 C:/Users/Vitor Castro/Desktop/grupo69/articles.h File Reference	17
4.2.1 Function Documentation	18
4.2.1.1 get_all_articles()	18
4.2.1.2 get_all_revisions()	19
4.2.1.3 get_article_timestamp()	19
4.2.1.4 get_article_title()	20
4.2.1.5 get_unique_articles()	20
4.2.1.6 getTitlesWithPrefix()	21
4.2.1.7 getTop20LargestArticles()	21
4.2.1.8 getTopNArticlesWithMoreWords()	22
4.2.1.9 onPageArticles()	22
4.3 C:/Users/Vitor Castro/Desktop/grupo69/hashtable.c File Reference	23
4.3.1 Function Documentation	24
4.3.1.1 clean_everything()	24
4.3.1.2 freeIterator()	24
4.3.1.3 getArticle()	24

4.3.1.4	getHashtableIterator()	25
4.3.1.5	getNextFromIterator()	25
4.3.1.6	getRevision()	25
4.3.1.7	getUser()	26
4.3.1.8	initHashtables()	26
4.3.1.9	insertOrUpdateArticle()	26
4.3.1.10	insertOrUpdateUser()	27
4.4	C:/Users/Vitor Castro/Desktop/grupo69/hashtable.h File Reference	28
4.4.1	Typedef Documentation	29
4.4.1.1	TAD_istruct	29
4.4.2	Function Documentation	29
4.4.2.1	clean_everything()	29
4.4.2.2	freeliterator()	29
4.4.2.3	getArticle()	29
4.4.2.4	getHashtableIterator()	30
4.4.2.5	getNextFromIterator()	30
4.4.2.6	getRevision()	30
4.4.2.7	getUser()	31
4.4.2.8	initHashtables()	31
4.4.2.9	insertOrUpdateArticle()	32
4.4.2.10	insertOrUpdateUser()	32
4.5	C:/Users/Vitor Castro/Desktop/grupo69/interface.c File Reference	33
4.5.1	Function Documentation	33
4.5.1.1	all_articles()	34
4.5.1.2	all_revisions()	34
4.5.1.3	article_timestamp()	34
4.5.1.4	article_title()	34
4.5.1.5	clean()	34
4.5.1.6	contributor_name()	35
4.5.1.7	init()	35

4.5.1.8	load()	35
4.5.1.9	titles_with_prefix()	35
4.5.1.10	top_10_contributors()	35
4.5.1.11	top_20_largest_articles()	36
4.5.1.12	top_N_articles_with_more_words()	36
4.5.1.13	unique_articles()	36
4.6	C:/Users/Vitor Castro/Desktop/grupo69/interface.h File Reference	36
4.6.1	Typedef Documentation	37
4.6.1.1	TAD_istruct	37
4.6.2	Function Documentation	37
4.6.2.1	all_articles()	37
4.6.2.2	all_revisions()	37
4.6.2.3	article_timestamp()	37
4.6.2.4	article_title()	38
4.6.2.5	clean()	38
4.6.2.6	contributor_name()	38
4.6.2.7	init()	38
4.6.2.8	load()	38
4.6.2.9	titles_with_prefix()	39
4.6.2.10	top_10_contributors()	39
4.6.2.11	top_20_largest_articles()	39
4.6.2.12	top_N_articles_with_more_words()	39
4.6.2.13	unique_articles()	39
4.7	C:/Users/Vitor Castro/Desktop/grupo69/parser.c File Reference	40
4.7.1	Function Documentation	40
4.7.1.1	parseWikiData()	40
4.8	C:/Users/Vitor Castro/Desktop/grupo69/parser.h File Reference	41
4.8.1	Typedef Documentation	41
4.8.1.1	TAD_istruct	41
4.8.2	Function Documentation	41

4.8.2.1	parseWikiData()	41
4.9	C:/Users/Vitor Castro/Desktop/grupo69/program.c File Reference	42
4.9.1	Function Documentation	42
4.9.1.1	main()	42
4.10	C:/Users/Vitor Castro/Desktop/grupo69/users.c File Reference	42
4.10.1	Function Documentation	43
4.10.1.1	getContributorName()	43
4.10.1.2	getTop10Contributors()	43
4.10.1.3	onPageUsers()	44
4.11	C:/Users/Vitor Castro/Desktop/grupo69/users.h File Reference	44
4.11.1	Function Documentation	45
4.11.1.1	getContributorName()	45
4.11.1.2	getTop10Contributors()	45
4.11.1.3	onPageUsers()	46
Index		47

Chapter 1

Data Structure Index

1.1 Data Structures

Here are the data structures with brief descriptions:

article	5
revision	6
TCD_istruct	7
user	8

Chapter 2

File Index

2.1 File List

Here is a list of all files with brief descriptions:

C:/Users/Vitor Castro/Desktop/grupo69/ articles.c	11
C:/Users/Vitor Castro/Desktop/grupo69/ articles.h	17
C:/Users/Vitor Castro/Desktop/grupo69/ hashtable.c	23
C:/Users/Vitor Castro/Desktop/grupo69/ hashtable.h	28
C:/Users/Vitor Castro/Desktop/grupo69/ interface.c	33
C:/Users/Vitor Castro/Desktop/grupo69/ interface.h	36
C:/Users/Vitor Castro/Desktop/grupo69/ parser.c	40
C:/Users/Vitor Castro/Desktop/grupo69/ parser.h	41
C:/Users/Vitor Castro/Desktop/grupo69/ program.c	42
C:/Users/Vitor Castro/Desktop/grupo69/ users.c	42
C:/Users/Vitor Castro/Desktop/grupo69/ users.h	44

Chapter 3

Data Structure Documentation

3.1 article Struct Reference

```
#include <hashtable.h>
```

Data Fields

- long [id](#)
- long [size](#)
- long [nWords](#)
- char * [title](#)
- GHashTable * [revisions](#)

3.1.1 Detailed Description

Definition at line 16 of file hashtable.h.

3.1.2 Field Documentation

3.1.2.1 id

```
long id
```

Definition at line 17 of file hashtable.h.

3.1.2.2 nWords

```
long nWords
```

Definition at line 17 of file hashtable.h.

3.1.2.3 revisions

```
GHashTable* revisions
```

Definition at line 19 of file hashtable.h.

3.1.2.4 size

```
long size
```

Definition at line 17 of file hashtable.h.

3.1.2.5 title

```
char* title
```

Definition at line 18 of file hashtable.h.

The documentation for this struct was generated from the following file:

- C:/Users/Vitor Castro/Desktop/grupo69/[hashtable.h](#)

3.2 revision Struct Reference

```
#include <hashtable.h>
```

Data Fields

- long [id](#)
- char * [timestamp](#)

3.2.1 Detailed Description

Definition at line 27 of file hashtable.h.

3.2.2 Field Documentation

3.2.2.1 id

```
long id
```

Definition at line 28 of file hashtable.h.

3.2.2.2 timestamp

```
char* timestamp
```

Definition at line 29 of file hashtable.h.

The documentation for this struct was generated from the following file:

- C:/Users/Vitor Castro/Desktop/grupo69/[hashtable.h](#)

3.3 TCD_istruct Struct Reference

```
#include <hashtable.h>
```

Data Fields

- GHashTable * [users](#)
- GHashTable * [articles](#)
- long [allArticles](#)
- long [uniqueArticles](#)
- long [allRevisions](#)

3.3.1 Detailed Description

Definition at line 8 of file hashtable.h.

3.3.2 Field Documentation

3.3.2.1 allArticles

```
long allArticles
```

Definition at line 11 of file hashtable.h.

3.3.2.2 allRevisions

```
long allRevisions
```

Definition at line 13 of file hashtable.h.

3.3.2.3 articles

```
GHashTable* articles
```

Definition at line 10 of file hashtable.h.

3.3.2.4 uniqueArticles

```
long uniqueArticles
```

Definition at line 12 of file hashtable.h.

3.3.2.5 users

```
GHashTable* users
```

Definition at line 9 of file hashtable.h.

The documentation for this struct was generated from the following file:

- C:/Users/Vitor Castro/Desktop/grupo69/[hashtable.h](#)

3.4 user Struct Reference

```
#include <hashtable.h>
```


Data Fields

- long [id](#)
- long [contributions](#)
- char * [username](#)

3.4.1 Detailed Description

Definition at line 22 of file hashtable.h.

3.4.2 Field Documentation

3.4.2.1 contributions

```
long contributions
```

Definition at line 23 of file hashtable.h.

3.4.2.2 id

```
long id
```

Definition at line 23 of file hashtable.h.

3.4.2.3 username

```
char* username
```

Definition at line 24 of file hashtable.h.

The documentation for this struct was generated from the following file:

- C:/Users/Vitor Castro/Desktop/grupo69/[hashtable.h](#)

Chapter 4

File Documentation

4.1 C:/Users/Vitor Castro/Desktop/grupo69/articles.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "hashtable.h"
#include "articles.h"
```

Functions

- long [wordCounter](#) (char *revisionText, long *sizeBytes)
Conta o número de palavras e o tamanho do texto.
- int [isTitlePrefix](#) (char *prefix, char *title)
Verifica que aquele título tem aquele prefixo.
- [TAD_istruct onPageArticles](#) ([TAD_istruct](#) qs, long articleId, char *title, char *revisionText, long revisionId, long revisionParentId, char *revisionTimestamp)
Função que trata os dados relativos a artigos, recebidos do parsing do XML, e os envia para a hashtable.
- long [get_all_articles](#) ([TAD_istruct](#) qs)
Função que obtém o número total de artigos analisados no decurso da execução.
- long [get_unique_articles](#) ([TAD_istruct](#) qs)
Função que obtém o número de artigos únicos existentes no decurso da execução.
- long [get_all_revisions](#) ([TAD_istruct](#) qs)
Função que obtém o número de revisões feitas no decurso da execução.
- char * [get_article_title](#) (long article_id, [TAD_istruct](#) qs)
Função que obtém o título de um artigo através do seu ID.
- char * [get_article_timestamp](#) (long article_id, long revision_id, [TAD_istruct](#) qs)
Função que obtém o timestamp de um artigo através do seu ID de artigo e de revisão.
- long * [getTop20LargestArticles](#) ([TAD_istruct](#) qs)
Função que obtém o top dos 20 maiores artigos.
- long * [getTopNArticlesWithMoreWords](#) (int n, [TAD_istruct](#) qs)
Função que obtém o top dos N artigos com mais palavras.
- char ** [getTitlesWithPrefix](#) (char *prefix, [TAD_istruct](#) qs)
Função que obtém o top dos N artigos com mais palavras.

4.1.1 Function Documentation

4.1.1.1 `get_all_articles()`

```
long get_all_articles (
    TAD_istruct qs )
```

Função que obtém o número total de artigos analisados no decurso da execução.

Retorna o seu campo `allArticles`.

Parameters

<i>qs</i>	A estrutura geral do programa.
-----------	--------------------------------

Returns

O número de artigos analisados.

Definition at line 155 of file `articles.c`.

4.1.1.2 `get_all_revisions()`

```
long get_all_revisions (
    TAD_istruct qs )
```

Função que obtém o número de revisões feitas no decurso da execução.

Retorna o seu campo `allRevisions`.

Parameters

<i>qs</i>	A estrutura geral do programa.
-----------	--------------------------------

Returns

O número de revisões.

Definition at line 185 of file `articles.c`.

4.1.1.3 `get_article_timestamp()`

```
char* get_article_timestamp (
    long article_id,
    long revision_id,
    TAD_istruct qs )
```

Função que obtém o timestamp de um artigo através do seu ID de artigo e de revisão.

Verifica se aquele artigo existe e, caso exista, retorna o seu campo timestamp.

See also

[getArticle\(\)](#)
[getRevision\(\)](#)

Parameters

<i>article_id</i>	O ID do artigo.
<i>revision↔ _id</i>	o ID da revisão.
<i>qs</i>	A estrutura geral do programa.

Returns

O timestamp do artigo ou `NULL`.

Definition at line 227 of file articles.c.

4.1.1.4 `get_article_title()`

```
char* get_article_title (
    long article_id,
    TAD_istruct qs )
```

Função que obtém o título de um artigo através do seu ID.

Verifica se aquele artigo existe e, caso exista, retorna o seu campo título.

See also

[getArticle\(\)](#)

Parameters

<i>article↔ _id</i>	O ID do artigo.
<i>qs</i>	A estrutura geral do programa.

Returns

O título do artigo ou `NULL`.

Definition at line 203 of file `articles.c`.

4.1.1.5 `get_unique_articles()`

```
long get_unique_articles (
    TAD_istruct qs )
```

Função que obtém o número de artigos únicos existentes no decurso da execução.

Retorna o seu campo `uniqueArticles`.

Parameters

<i>qs</i>	A estrutura geral do programa.
-----------	--------------------------------

Returns

O número de artigos únicos.

Definition at line 170 of file `articles.c`.

4.1.1.6 `getTitlesWithPrefix()`

```
char** getTitlesWithPrefix (
    char * prefix,
    TAD_istruct qs )
```

Função que obtém o top dos N artigos com mais palavras.

Cria um iterador da hashtable correspondente aos artigos. Itera pela hashtable e verifica através da função `isTitlePrefix` se aquele título tem aquele prefixo. Caso tenha, aumenta o array e guarda o título. No fim, é ordenado o array.

See also

[isTitlePrefix\(\)](#)
[getHashtableIterator\(\)](#)
[getNextFromIterator\(\)](#)

Parameters

<i>prefix</i>	O prefixo a ser pesquisado.
<i>qs</i>	A estrutura geral do programa.

Returns

Um array com os títulos dos artigos com aquele prefixo, ou NULL, caso não hajam.

Definition at line 489 of file articles.c.

4.1.1.7 `getTop20LargestArticles()`

```
long* getTop20LargestArticles (
    TAD_istruct qs )
```

Função que obtém o top dos 20 maiores artigos.

Cria um iterador da hashtable correspondente aos artigos. Cria e inicializa um array de 20 artigos. Itera pela hashtable e compara o tamanho do artigo a ser iterado com o top, começando pelo último lugar deste. Caso seja maior ou igual que o último, continua a fazer comparações com os acima. Caso o tamanho dos artigos forem iguais, é feita a decisão pelo ID destes. Caso entre no top, é feito o reajuste dos lugares abaixo.

See also

[getHashtableIterator\(\)](#)
[getNextFromIterator\(\)](#)

Parameters

<code>qs</code>	A estrutura geral do programa.
-----------------	--------------------------------

Returns

Um array com os IDs dos 20 maiores artigos.

Definition at line 258 of file articles.c.

4.1.1.8 `getTopNArticlesWithMoreWords()`

```
long* getTopNArticlesWithMoreWords (
    int n,
    TAD_istruct qs )
```

Função que obtém o top dos N artigos com mais palavras.

Cria um iterador da hashtable correspondente aos artigos. Cria e inicializa um array de N artigos. Itera pela hashtable e compara o número de palavras do artigo a ser iterado com o top, começando pelo último lugar deste. Caso seja maior ou igual que o último, continua a fazer comparações com os acima. Caso o número de palavras dos artigos forem iguais, é feita a decisão pelo ID destes. Caso entre no top, é feito o reajuste dos lugares abaixo.

See also

[getHashtableIterator\(\)](#)
[getNextFromIterator\(\)](#)

Parameters

<i>n</i>	O tamanho do top.
<i>qs</i>	A estrutura geral do programa.

Returns

Um array com os IDs dos N artigos com mais palavras.

Definition at line 376 of file articles.c.

4.1.1.9 isTitlePrefix()

```
int isTitlePrefix (
    char * prefix,
    char * title )
```

Verifica que aquele título tem aquele prefixo.

Compara char a char, tendo em conta também o tamanho, com vista à determinação pedida.

Parameters

<i>prefix</i>	O prefixo em conta.
<i>title</i>	O título a comparar.

Returns

O valor 1 caso seja e 0 caso não seja prefixo.

Definition at line 51 of file articles.c.

4.1.1.10 onPageArticles()

```
TAD_istruct onPageArticles (
    TAD_istruct qs,
    long articleId,
    char * title,
    char * revisionText,
    long revisionId,
    long revisionParentId,
    char * revisionTimestamp )
```

Função que trata os dados relativos a artigos, recebidos do parsing do XML, e os envia para a hashtable.

Obtém informação sobre número de palavras e sobre tamanho do artigo recebido através da função wordCounter e chama as funções de update da hashtable, com as informações que obteve. Dependendo de ter atualizado ou não determinados campos da hashtable, atualiza os diversos contadores de articles.

See also

[wordCounter\(\)](#)
[insertOrUpdateArticle\(\)](#)

Parameters

<i>qs</i>	A estrutura geral do programa.
<i>articleId</i>	A ID do artigo.
<i>title</i>	O título do artigo.
<i>revisionText</i>	O conteúdo do artigo.
<i>revisionId</i>	O ID da revisão.
<i>revisionParentId</i>	O ID da revisão pai.
<i>revisionTimestamp</i>	O timestamp da revisão.

Returns

A estrutura geral atualizada.

Definition at line 99 of file articles.c.

4.1.1.11 wordCounter()

```
long wordCounter (
    char * revisionText,
    long * sizeBytes )
```

Conta o número de palavras e o tamanho do texto.

Utiliza flags para contar palavras e ignorar espaços entre elas e caracteres especiais. Atualiza o valor de sizeBytes para não se perder tempo com strlen, ou seja, é 2 em 1.

Parameters

<i>revisionText</i>	O texto a analisar, correspondente ao conteúdo do artigo.
<i>sizeBytes</i>	Apontador a atualizar para dar o tamanho do artigo.

Returns

O número de palavras do artigo.

Definition at line 19 of file articles.c.

4.2 C:/Users/Vitor Castro/Desktop/grupo69/articles.h File Reference

```
#include "hashtable.h"
```

Functions

- [TAD_istruct onPageArticles](#) ([TAD_istruct](#) qs, long articleId, char *title, char *revisionText, long revisionId, long revisionParentId, char *revisionTimestamp)
Função que trata os dados relativos a artigos, recebidos do parsing do XML, e os envia para a hashtable.
- long [get_all_articles](#) ([TAD_istruct](#) qs)
Função que obtém o número total de artigos analisados no decurso da execução.
- long [get_unique_articles](#) ([TAD_istruct](#) qs)
Função que obtém o número de artigos únicos existentes no decurso da execução.
- long [get_all_revisions](#) ([TAD_istruct](#) qs)
Função que obtém o número de revisões feitas no decurso da execução.
- char * [get_article_title](#) (long article_id, [TAD_istruct](#) qs)
Função que obtém o título de um artigo através do seu ID.
- char * [get_article_timestamp](#) (long article_id, long revision_id, [TAD_istruct](#) qs)
Função que obtém o timestamp de um artigo através do seu ID de artigo e de revisão.
- long * [getTop20LargestArticles](#) ([TAD_istruct](#) qs)
Função que obtém o top dos 20 maiores artigos.
- long * [getTopNArticlesWithMoreWords](#) (int n, [TAD_istruct](#) qs)
Função que obtém o top dos N artigos com mais palavras.
- char ** [getTitlesWithPrefix](#) (char *prefix, [TAD_istruct](#) qs)
Função que obtém o top dos N artigos com mais palavras.

4.2.1 Function Documentation

4.2.1.1 [get_all_articles\(\)](#)

```
long get_all_articles (
    TAD\_istruct qs )
```

Função que obtém o número total de artigos analisados no decurso da execução.

Retorna o seu campo allArticles.

Parameters

<i>qs</i>	A estrutura geral do programa.
-----------	--------------------------------

Returns

O número de artigos analisados.

Definition at line 155 of file articles.c.

4.2.1.2 `get_all_revisions()`

```
long get_all_revisions (
    TAD_istruct qs )
```

Função que obtém o número de revisões feitas no decurso da execução.

Retorna o seu campo `allRevisions`.

Parameters

<i>qs</i>	A estrutura geral do programa.
-----------	--------------------------------

Returns

O número de revisões.

Definition at line 185 of file `articles.c`.

4.2.1.3 `get_article_timestamp()`

```
char* get_article_timestamp (
    long article_id,
    long revision_id,
    TAD_istruct qs )
```

Função que obtém o timestamp de um artigo através do seu ID de artigo e de revisão.

Verifica se aquele artigo existe e, caso exista, retorna o seu campo `timestamp`.

See also

[getArticle\(\)](#)
[getRevision\(\)](#)

Parameters

<i>article_id</i>	O ID do artigo.
<i>revision↔_id</i>	o ID da revisão.
<i>qs</i>	A estrutura geral do programa.

Returns

O timestamp do artigo ou `NULL`.

Definition at line 227 of file `articles.c`.

4.2.1.4 `get_article_title()`

```
char* get_article_title (
    long article_id,
    TAD_istruct qs )
```

Função que obtém o título de um artigo através do seu ID.

Verifica se aquele artigo existe e, caso exista, retorna o seu campo título.

See also

[getArticle\(\)](#)

Parameters

<i>article_id</i>	O ID do artigo.
<i>qs</i>	A estrutura geral do programa.

Returns

O título do artigo ou `NULL`.

Definition at line 203 of file `articles.c`.

4.2.1.5 `get_unique_articles()`

```
long get_unique_articles (
    TAD_istruct qs )
```

Função que obtém o número de artigos únicos existentes no decurso da execução.

Retorna o seu campo `uniqueArticles`.

Parameters

<i>qs</i>	A estrutura geral do programa.
-----------	--------------------------------

Returns

O número de artigos únicos.

Definition at line 170 of file `articles.c`.

4.2.1.6 `getTitlesWithPrefix()`

```
char** getTitlesWithPrefix (
    char * prefix,
    TAD_istruct qs )
```

Função que obtém o top dos N artigos com mais palavras.

Cria um iterador da hashtable correspondente aos artigos. Itera pela hashtable e verifica através da função `isTitlePrefix` se aquele título tem aquele prefixo. Caso tenha, aumenta o array e guarda o título. No fim, é ordenado o array.

See also

[isTitlePrefix\(\)](#)
[getHashtableIterator\(\)](#)
[getNextFromIterator\(\)](#)

Parameters

<i>prefix</i>	O prefixo a ser pesquisado.
<i>qs</i>	A estrutura geral do programa.

Returns

Um array com os títulos dos artigos com aquele prefixo, ou NULL, caso não hajam.

Definition at line 489 of file `articles.c`.

4.2.1.7 `getTop20LargestArticles()`

```
long* getTop20LargestArticles (
    TAD_istruct qs )
```

Função que obtém o top dos 20 maiores artigos.

Cria um iterador da hashtable correspondente aos artigos. Cria e inicializa um array de 20 artigos. Itera pela hashtable e compara o tamanho do artigo a ser iterado com o top, começando pelo último lugar deste. Caso seja maior ou igual que o último, continua a fazer comparações com os acima. Caso o tamanho dos artigos forem iguais, é feita a decisão pelo ID destes. Caso entre no top, é feito o reajuste dos lugares abaixo.

See also

[getHashtableIterator\(\)](#)
[getNextFromIterator\(\)](#)

Parameters

<i>qs</i>	A estrutura geral do programa.
-----------	--------------------------------

Returns

Um array com os IDs dos 20 maiores artigos.

Definition at line 258 of file articles.c.

4.2.1.8 `getTopNArticlesWithMoreWords()`

```
long* getTopNArticlesWithMoreWords (
    int n,
    TAD_istruct qs )
```

Função que obtém o top dos N artigos com mais palavras.

Cria um iterador da hashtable correspondente aos artigos. Cria e inicializa um array de N artigos. Itera pela hashtable e compara o número de palavras do artigo a ser iterado com o top, começando pelo último lugar deste. Caso seja maior ou igual que o último, continua a fazer comparações com os acima. Caso o número de palavras dos artigos forem iguais, é feita a decisão pelo ID destes. Caso entre no top, é feito o reajuste dos lugares abaixo.

See also

[getHashtableIterator\(\)](#)
[getNextFromIterator\(\)](#)

Parameters

<i>n</i>	O tamanho do top.
<i>qs</i>	A estrutura geral do programa.

Returns

Um array com os IDs dos N artigos com mais palavras.

Definition at line 376 of file articles.c.

4.2.1.9 `onPageArticles()`

```
TAD_istruct onPageArticles (
    TAD_istruct qs,
    long articleId,
    char * title,
    char * revisionText,
    long revisionId,
    long revisionParentId,
    char * revisionTimestamp )
```

Função que trata os dados relativos a artigos, recebidos do parsing do XML, e os envia para a hashtable.

Obtém informação sobre número de palavras e sobre tamanho do artigo recebido através da função wordCounter e chama as funções de update da hashtable, com as informações que obteve. Dependendo de ter atualizado ou não determinados campos da hashtable, atualiza os diversos contadores de articles.

See also

[wordCounter\(\)](#)
[insertOrUpdateArticle\(\)](#)

Parameters

<i>qs</i>	A estrutura geral do programa.
<i>articleId</i>	A ID do artigo.
<i>title</i>	O título do artigo.
<i>revisionText</i>	O conteúdo do artigo.
<i>revisionId</i>	O ID da revisão.
<i>revisionParentId</i>	O ID da revisão pai.
<i>revisionTimestamp</i>	O timestamp da revisão.

Returns

A estrutura geral atualizada.

Definition at line 99 of file articles.c.

4.3 C:/Users/Vitor Castro/Desktop/grupo69/hashtable.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <glib.h>
#include "hashtable.h"
```

Functions

- [TAD_istruct initHashtables](#) (TAD_istruct qs)
Inicia a estrutura geral do programa.
- [TAD_istruct insertOrUpdateUser](#) (TAD_istruct qs, long id, char *username, int *userWasFound)
Insere ou atualiza um utilizador na hashtable de utilizadores.
- [TAD_istruct insertOrUpdateArticle](#) (TAD_istruct qs, long id, char *title, long revisionId, char *revisionTimestamp, long sizeBytes, long nWords, int *articleFound, int *articleUpdated)
Insere ou atualiza um artigo na hashtable de utilizadores.
- [TAD_istruct clean_everything](#) (TAD_istruct qs)
Limpa a estrutura geral do programa.
- struct [user](#) * [getUser](#) (TAD_istruct qs, long id)
Obtém um utilizador de acordo com a ID.
- struct [article](#) * [getArticle](#) (TAD_istruct qs, long id)
Obtém um artigo de acordo com a ID.
- struct [revision](#) * [getRevision](#) (void *revisionsHashTable, long revisionId)
Obtém uma revisão de acordo com a ID.
- void * [getHashtableIterator](#) (GHashTable *hashtable)
Obtém um iterador.
- int [getNextFromIterator](#) (void *iterator, void *key, void *value)
Obtém o próximo elemento a partir de um iterador.
- void [freeliterator](#) (void *iterator)
Liberta a memória de um iterador.

4.3.1 Function Documentation

4.3.1.1 `clean_everything()`

```
TAD_istruct clean_everything (
    TAD_istruct qs )
```

Limpa a estrutura geral do programa.

Parameters

<i>qs</i>	A estrutura geral do programa.
-----------	--------------------------------

Returns

A estrutura vazia.

Definition at line 214 of file hashtable.c.

4.3.1.2 `freeIterator()`

```
void freeIterator (
    void * iterator )
```

Liberta a memória de um iterador.

Definition at line 298 of file hashtable.c.

4.3.1.3 `getArticle()`

```
struct article* getArticle (
    TAD_istruct qs,
    long id )
```

Obtém um artigo de acordo com a ID.

Parameters

<i>qs</i>	A hashtable.
<i>id</i>	O ID do artigo a obter.

Returns

O artigo.

Definition at line 256 of file hashtable.c.

4.3.1.4 getHashtableIterator()

```
void* getHashtableIterator (
    GHashTable * hashtable )
```

Obtém um iterador.

Definition at line 276 of file hashtable.c.

4.3.1.5 getNextFromIterator()

```
int getNextFromIterator (
    void * iterator,
    void * key,
    void * value )
```

Obtém o próximo elemento a partir de um iterador.

Definition at line 288 of file hashtable.c.

4.3.1.6 getRevision()

```
struct revision* getRevision (
    void * revisionsHashTable,
    long revisionId )
```

Obtém uma revisão de acordo com a ID.

Parameters

<i>revisionsHashTable</i>	A hashtable de revisões.
<i>revisionId</i>	O ID da revisão a obter.

Returns

A revisão.

Definition at line 268 of file hashtable.c.

4.3.1.7 getUser()

```
struct user* getUser (
    TAD_istruct qs,
    long id )
```

Obtém um utilizador de acordo com a ID.

Parameters

<i>qs</i>	A hashtable.
<i>id</i>	O ID do utilizador a obter.

Returns

O utilizador.

Definition at line 244 of file hashtable.c.

4.3.1.8 initHashtables()

```
TAD_istruct initHashtables (
    TAD_istruct qs )
```

Inicia a estrutura geral do programa.

Parameters

<i>qs</i>	A estrutura geral do programa que é nula ainda.
-----------	---

Returns

A estrutura iniciada.

Definition at line 13 of file hashtable.c.

4.3.1.9 insertOrUpdateArticle()

```
TAD_istruct insertOrUpdateArticle (
    TAD_istruct qs,
    long id,
    char * title,
    long revisionId,
    char * revisionTimestamp,
    long sizeBytes,
```

```

    long nWords,
    int * articleFound,
    int * articleUpdated )

```

Insere ou atualiza um artigo na hashtable de utilizadores.

Copiar strings porque o parser pode apagá-las e a hash table só guarda um pointer. Copiar longs usadas como key também, porque keys são usadas a partir dos seus pointers. Criar nova revisão (necessária caso o artigo já exista ou não). Informar se o artigo foi encontrado ou não através de articleFound e articleUpdated. Criar ou atualizar o artigo dependendo se este foi encontrado ou não.

Parameters

<i>qs</i>	A estrutura geral do programa.
<i>id</i>	A ID do artigo.
<i>title</i>	O título do artigo.
<i>revisionId</i>	O ID da revisão.
<i>revisionTimestamp</i>	O timestamp da revisão.
<i>sizeBytes</i>	O tamanho do artigo.
<i>nWords</i>	O número de palavras do artigo.
<i>articleFound</i>	Passado para fazer controlo nos contadores de articles.
<i>articleUpdated</i>	Passado para fazer controlo nos contadores de articles.

Returns

A estrutura atualizada.

Definition at line 105 of file hashtable.c.

4.3.1.10 insertOrUpdateUser()

```

TAD_istruct insertOrUpdateUser (
    TAD_istruct qs,
    long id,
    char * username,
    int * userWasFound )

```

Insere ou atualiza um utilizador na hashtable de utilizadores.

Procurar pelo utilizador na hash table e, caso o utilizador não foi encontrado, criar e inserir novo utilizador. Caso tenha sido encontrado, apenas aumentar uma contribuição ao utilizador.

Parameters

<i>qs</i>	A estrutura geral do programa.
<i>id</i>	A ID do contribuidor.
<i>username</i>	O username do contribuidor.
<i>userWasFound</i>	Passado para fazer controlo.

Returns

A estrutura atualizada.

Definition at line 37 of file hashtable.c.

4.4 C:/Users/Vitor Castro/Desktop/grupo69/hashtable.h File Reference

```
#include <glib.h>
```

Data Structures

- struct [TCD_istruct](#)
- struct [article](#)
- struct [user](#)
- struct [revision](#)

Typedefs

- typedef struct [TCD_istruct](#) * [TAD_istruct](#)

Functions

- [TAD_istruct initHashtables](#) ([TAD_istruct](#) qs)
Inicia a estrutura geral do programa.
- [TAD_istruct insertOrUpdateUser](#) ([TAD_istruct](#) qs, long id, char *username, int *userWasFound)
Insere ou atualiza um utilizador na hashtable de utilizadores.
- [TAD_istruct insertOrUpdateArticle](#) ([TAD_istruct](#) qs, long id, char *title, long revisionId, char *revisionTimestamp, long sizeBytes, long nWords, int *articleFound, int *articleUpdated)
Insere ou atualiza um artigo na hashtable de utilizadores.
- [TAD_istruct clean_everything](#) ([TAD_istruct](#) qs)
Limpa a estrutura geral do programa.
- struct [user](#) * [getUser](#) ([TAD_istruct](#) qs, long id)
Obtém um utilizador de acordo com a ID.
- struct [article](#) * [getArticle](#) ([TAD_istruct](#) qs, long id)
Obtém um artigo de acordo com a ID.
- struct [revision](#) * [getRevision](#) (void *revisionsHashTable, long revisionId)
Obtém uma revisão de acordo com a ID.
- void * [getHashtableIterator](#) (GHashTable *hashtable)
Obtém um iterador.
- int [getNextFromIterator](#) (void *iterator, void *key, void *value)
Obtém o próximo elemento a partir de um iterador.
- void [freeliterator](#) (void *iterator)
Liberta a memória de um iterador.

4.4.1 Typedef Documentation

4.4.1.1 TAD_istruct

```
typedef struct TCD_istruct* TAD_istruct
```

Definition at line 6 of file hashtable.h.

4.4.2 Function Documentation

4.4.2.1 clean_everything()

```
TAD_istruct clean_everything (
    TAD_istruct qs )
```

Limpa a estrutura geral do programa.

Parameters

<i>qs</i>	A estrutura geral do programa.
-----------	--------------------------------

Returns

A estrutura vazia.

Definition at line 214 of file hashtable.c.

4.4.2.2 freeIterator()

```
void freeIterator (
    void * iterator )
```

Liberta a memória de um iterador.

Definition at line 298 of file hashtable.c.

4.4.2.3 getArticle()

```
struct article* getArticle (
    TAD_istruct qs,
    long id )
```

Obtém um artigo de acordo com a ID.

Parameters

<i>qs</i>	A hashtable.
<i>id</i>	O ID do artigo a obter.

Returns

O artigo.

Definition at line 256 of file hashtable.c.

4.4.2.4 getHashtableIterator()

```
void* getHashtableIterator (
    GHashTable * hashtable )
```

Obtém um iterador.

Definition at line 276 of file hashtable.c.

4.4.2.5 getNextFromIterator()

```
int getNextFromIterator (
    void * iterator,
    void * key,
    void * value )
```

Obtém o próximo elemento a partir de um iterador.

Definition at line 288 of file hashtable.c.

4.4.2.6 getRevision()

```
struct revision* getRevision (
    void * revisionsHashTable,
    long revisionId )
```

Obtém uma revisão de acordo com a ID.

Parameters

<i>revisionsHashTable</i>	A hashtable de revisões.
<i>revisionId</i>	O ID da revisão a obter.

Returns

A revisão.

Definition at line 268 of file hashtable.c.

4.4.2.7 getUser()

```
struct user* getUser (
    TAD_istruct qs,
    long id )
```

Obtém um utilizador de acordo com a ID.

Parameters

<i>qs</i>	A hashtable.
<i>id</i>	O ID do utilizador a obter.

Returns

O utilizador.

Definition at line 244 of file hashtable.c.

4.4.2.8 initHashtables()

```
TAD_istruct initHashtables (
    TAD_istruct qs )
```

Inicia a estrutura geral do programa.

Parameters

<i>qs</i>	A estrutura geral do programa que é nula ainda.
-----------	---

Returns

A estrutura iniciada.

Definition at line 13 of file hashtable.c.

4.4.2.9 insertOrUpdateArticle()

```
TAD_istruct insertOrUpdateArticle (
    TAD_istruct qs,
    long id,
    char * title,
    long revisionId,
    char * revisionTimestamp,
    long sizeBytes,
    long nWords,
    int * articleFound,
    int * articleUpdated )
```

Insere ou atualiza um artigo na hashtable de utilizadores.

Copiar strings porque o parser pode apagá-las e a hash table só guarda um pointer. Copiar longs usadas como key também, porque keys são usadas a partir dos seus pointers. Criar nova revisão (necessária caso o artigo já exista ou não). Informar se o artigo foi encontrado ou não através de articleFound e articleUpdated. Criar ou atualizar o artigo dependendo se este foi encontrado ou não.

Parameters

<i>qs</i>	A estrutura geral do programa.
<i>id</i>	A ID do artigo.
<i>title</i>	O título do artigo.
<i>revisionId</i>	O ID da revisão.
<i>revisionTimestamp</i>	O timestamp da revisão.
<i>sizeBytes</i>	O tamanho do artigo.
<i>nWords</i>	O número de palavras do artigo.
<i>articleFound</i>	Passado para fazer controlo nos contadores de articles.
<i>articleUpdated</i>	Passado para fazer controlo nos contadores de articles.

Returns

A estrutura atualizada.

Definition at line 105 of file hashtable.c.

4.4.2.10 insertOrUpdateUser()

```
TAD_istruct insertOrUpdateUser (
    TAD_istruct qs,
    long id,
    char * username,
    int * userWasFound )
```

Insere ou atualiza um utilizador na hashtable de utilizadores.

Procurar pelo utilizador na hash table e, caso o utilizador não foi encontrado, criar e inserir novo utilizador. Caso tenha sido encontrado, apenas aumentar uma contribuição ao utilizador.

Parameters

<i>qs</i>	A estrutura geral do programa.
<i>id</i>	A ID do contribuidor.
<i>username</i>	O username do contribuidor.
<i>userWasFound</i>	Passado para fazer controlo.

Returns

A estrutura atualizada.

Definition at line 37 of file hashtable.c.

4.5 C:/Users/Vitor Castro/Desktop/grupo69/interface.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <glib.h>
#include "parser.h"
#include "hashtable.h"
#include "users.h"
#include "articles.h"
#include "interface.h"
```

Functions

- [TAD_istruct init](#) ()
- [TAD_istruct load](#) (TAD_istruct qs, int nsnaps, char *snaps_paths[])
- [TAD_istruct clean](#) (TAD_istruct qs)
- long [all_articles](#) (TAD_istruct qs)
- long [unique_articles](#) (TAD_istruct qs)
- long [all_revisions](#) (TAD_istruct qs)
- long * [top_10_contributors](#) (TAD_istruct qs)
- char * [contributor_name](#) (long contributor_id, TAD_istruct qs)
- long * [top_20_largest_articles](#) (TAD_istruct qs)
- char * [article_title](#) (long article_id, TAD_istruct qs)
- long * [top_N_articles_with_more_words](#) (int n, TAD_istruct qs)
- char ** [titles_with_prefix](#) (char *prefix, TAD_istruct qs)
- char * [article_timestamp](#) (long article_id, long revision_id, TAD_istruct qs)

4.5.1 Function Documentation

4.5.1.1 all_articles()

```
long all_articles (
    TAD_istruct qs )
```

Definition at line 49 of file interface.c.

4.5.1.2 all_revisions()

```
long all_revisions (
    TAD_istruct qs )
```

Definition at line 59 of file interface.c.

4.5.1.3 article_timestamp()

```
char* article_timestamp (
    long article_id,
    long revision_id,
    TAD_istruct qs )
```

Definition at line 95 of file interface.c.

4.5.1.4 article_title()

```
char* article_title (
    long article_id,
    TAD_istruct qs )
```

Definition at line 80 of file interface.c.

4.5.1.5 clean()

```
TAD_istruct clean (
    TAD_istruct qs )
```

Definition at line 44 of file interface.c.

4.5.1.6 contributor_name()

```
char* contributor_name (
    long contributor_id,
    TAD_istruct qs )
```

Definition at line 70 of file interface.c.

4.5.1.7 init()

```
TAD_istruct init ( )
```

Definition at line 11 of file interface.c.

4.5.1.8 load()

```
TAD_istruct load (
    TAD_istruct qs,
    int nsnaps,
    char * snaps_paths[] )
```

Definition at line 28 of file interface.c.

4.5.1.9 titles_with_prefix()

```
char** titles_with_prefix (
    char * prefix,
    TAD_istruct qs )
```

Definition at line 90 of file interface.c.

4.5.1.10 top_10_contributors()

```
long* top_10_contributors (
    TAD_istruct qs )
```

Definition at line 65 of file interface.c.

4.5.1.11 top_20_largest_articles()

```
long* top_20_largest_articles (
    TAD_istruct qs )
```

Definition at line 75 of file interface.c.

4.5.1.12 top_N_articles_with_more_words()

```
long* top_N_articles_with_more_words (
    int n,
    TAD_istruct qs )
```

Definition at line 85 of file interface.c.

4.5.1.13 unique_articles()

```
long unique_articles (
    TAD_istruct qs )
```

Definition at line 54 of file interface.c.

4.6 C:/Users/Vitor Castro/Desktop/grupo69/interface.h File Reference

Typedefs

- typedef struct TCD_istruct * TAD_istruct

Functions

- TAD_istruct init ()
- TAD_istruct load (TAD_istruct qs, int nsnaps, char *snaps_paths[])
- long all_articles (TAD_istruct qs)
- long unique_articles (TAD_istruct qs)
- long all_revisions (TAD_istruct qs)
- long * top_10_contributors (TAD_istruct qs)
- char * contributor_name (long contributor_id, TAD_istruct qs)
- long * top_20_largest_articles (TAD_istruct qs)
- char * article_title (long article_id, TAD_istruct qs)
- long * top_N_articles_with_more_words (int n, TAD_istruct qs)
- char ** titles_with_prefix (char *prefix, TAD_istruct qs)
- char * article_timestamp (long article_id, long revision_id, TAD_istruct qs)
- TAD_istruct clean (TAD_istruct qs)

4.6.1 Typedef Documentation

4.6.1.1 TAD_istruct

```
typedef struct TCD_istruct* TAD_istruct
```

Definition at line 4 of file interface.h.

4.6.2 Function Documentation

4.6.2.1 all_articles()

```
long all_articles (
    TAD_istruct qs )
```

Definition at line 49 of file interface.c.

4.6.2.2 all_revisions()

```
long all_revisions (
    TAD_istruct qs )
```

Definition at line 59 of file interface.c.

4.6.2.3 article_timestamp()

```
char* article_timestamp (
    long article_id,
    long revision_id,
    TAD_istruct qs )
```

Definition at line 95 of file interface.c.

4.6.2.4 article_title()

```
char* article_title (
    long article_id,
    TAD_istruct qs )
```

Definition at line 80 of file interface.c.

4.6.2.5 clean()

```
TAD_istruct clean (
    TAD_istruct qs )
```

Definition at line 44 of file interface.c.

4.6.2.6 contributor_name()

```
char* contributor_name (
    long contributor_id,
    TAD_istruct qs )
```

Definition at line 70 of file interface.c.

4.6.2.7 init()

```
TAD_istruct init ( )
```

Definition at line 11 of file interface.c.

4.6.2.8 load()

```
TAD_istruct load (
    TAD_istruct qs,
    int nsnaps,
    char * snaps_paths[ ] )
```

Definition at line 28 of file interface.c.

4.6.2.9 titles_with_prefix()

```
char** titles_with_prefix (  
    char * prefix,  
    TAD_istruct qs )
```

Definition at line 90 of file interface.c.

4.6.2.10 top_10_contributors()

```
long* top_10_contributors (  
    TAD_istruct qs )
```

Definition at line 65 of file interface.c.

4.6.2.11 top_20_largest_articles()

```
long* top_20_largest_articles (  
    TAD_istruct qs )
```

Definition at line 75 of file interface.c.

4.6.2.12 top_N_articles_with_more_words()

```
long* top_N_articles_with_more_words (  
    int n,  
    TAD_istruct qs )
```

Definition at line 85 of file interface.c.

4.6.2.13 unique_articles()

```
long unique_articles (  
    TAD_istruct qs )
```

Definition at line 54 of file interface.c.

4.7 C:/Users/Vitor Castro/Desktop/grupo69/parser.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <libxml/tree.h>
#include <libxml/parser.h>
#include "articles.h"
#include "users.h"
#include "parser.h"
```

Functions

- [TAD_istruct parseWikiData](#) ([TAD_istruct](#) qs, const char *docname)

Função que carrega e distribui as pages.

4.7.1 Function Documentation

4.7.1.1 parseWikiData()

```
TAD_istruct parseWikiData (
    TAD_istruct qs,
    const char * docname )
```

Função que carrega e distribui as pages.

Verifica o correto carregamento do ficheiro e obtém a root, percorrendo após isso as pages e dando-as à função `parsePage` para esta recolher dados.

See also

`parsePage()`

Parameters

<i>qs</i>	A estrutura geral do programa.
<i>docname</i>	O nome do documento a carregar.

Returns

A estrutura carregada.

Definition at line 188 of file `parser.c`.

4.8 C:/Users/Vitor Castro/Desktop/grupo69/parser.h File Reference

Typedefs

- typedef struct [TCD_istruct](#) * [TAD_istruct](#)

Functions

- [TAD_istruct](#) [parseWikiData](#) ([TAD_istruct](#) qs, const char *docname)
Função que carrega e distribui as pages.

4.8.1 Typedef Documentation

4.8.1.1 TAD_istruct

```
typedef struct TCD\_istruct* TAD\_istruct
```

Definition at line 4 of file parser.h.

4.8.2 Function Documentation

4.8.2.1 parseWikiData()

```
TAD\_istruct parseWikiData (  
    TAD\_istruct qs,  
    const char * docname )
```

Função que carrega e distribui as pages.

Verifica o correto carregamento do ficheiro e obtém a root, percorrendo após isso as pages e dando-as à função [parsePage](#) para esta recolher dados.

See also

[parsePage\(\)](#)

Parameters

<i>qs</i>	A estrutura geral do programa.
<i>docname</i>	O nome do documento a carregar.

Returns

A estrutura carregada.

Definition at line 188 of file parser.c.

4.9 C:/Users/Vitor Castro/Desktop/grupo69/program.c File Reference

```
#include <stdlib.h>
#include <stdio.h>
#include <time.h>
#include "interface.h"
```

Functions

- int [main](#) (int argc, char const *argv[])

4.9.1 Function Documentation

4.9.1.1 main()

```
int main (
    int argc,
    char const * argv[] )
```

Definition at line 103 of file program.c.

4.10 C:/Users/Vitor Castro/Desktop/grupo69/users.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "hashtable.h"
#include "users.h"
```

Functions

- [TAD_istruct onPageUsers](#) ([TAD_istruct](#) qs, long id, char *username, long articleId, long revisionId)
Função que trata os dados relativos a users, recebidos do parsing do XML, e os envia para a hashtable.
- char * [getContributorName](#) ([TAD_istruct](#) qs, long id)
Função que obtém o nome de um contribuidor através do seu ID.
- long * [getTop10Contributors](#) ([TAD_istruct](#) qs)
Função que obtém o top dos 10 melhores contribuidores.

4.10.1 Function Documentation

4.10.1.1 `getContributorName()`

```
char* getContributorName (
    TAD_istruct qs,
    long id )
```

Função que obtém o nome de um contribuidor através do seu ID.

Verifica se aquele contribuidor existe e, caso exista, retorna o seu campo username.

See also

[getUser\(\)](#)

Parameters

<i>qs</i>	A estrutura geral do programa.
<i>id</i>	O ID do contribuidor a pesquisar.

Returns

O username do contribuidor ou NULL.

Definition at line 50 of file users.c.

4.10.1.2 `getTop10Contributors()`

```
long* getTop10Contributors (
    TAD_istruct qs )
```

Função que obtém o top dos 10 melhores contribuidores.

Cria um iterador da hashtable correspondente aos utilizadores. Cria e inicializa um array de 10 utilizadores. Itera pela hashtable e compara o número de contribuições do utilizador a ser iterado com o top, começando pelo último lugar deste. Caso seja maior ou igual que o último, continua a fazer comparações com os acima. Caso o número de contribuições entre dois utilizadores forem iguais, é feita a decisão pelo ID destes. Caso entre no top, é feito o reajuste dos lugares abaixo.

See also

[getHashtableIterator\(\)](#)

[getNextFromIterator\(\)](#)

Parameters

<i>qs</i>	A estrutura geral do programa.
-----------	--------------------------------

Returns

Um array com os IDs dos 10 melhores contribuidores.

Definition at line 79 of file users.c.

4.10.1.3 onPageUsers()

```
TAD_istruct onPageUsers (
    TAD_istruct qs,
    long id,
    char * username,
    long articleId,
    long revisionId )
```

Função que trata os dados relativos a users, recebidos do parsing do XML, e os envia para a hashtable.

Verifica se aquele artigo existe e se aquela revisão também já existe e, nesse caso, não modifica nada. Caso contrário, chama as funções de update da hashtable.

Parameters

<i>qs</i>	A estrutura geral do programa.
<i>id</i>	A ID do contribuidor da revisão.
<i>username</i>	O username do contribuidor da revisão.
<i>articleId</i>	O ID do artigo.
<i>revisionId</i>	O ID da revisão.

Returns

A estrutura geral atualizada.

Definition at line 23 of file users.c.

4.11 C:/Users/Vitor Castro/Desktop/grupo69/users.h File Reference

Functions

- [TAD_istruct onPageUsers](#) ([TAD_istruct](#) qs, long id, char *username, long articleId, long revisionId)
Função que trata os dados relativos a users, recebidos do parsing do XML, e os envia para a hashtable.
- char * [getContributorName](#) ([TAD_istruct](#) qs, long id)
Função que obtém o nome de um contribuidor através do seu ID.
- long * [getTop10Contributors](#) ([TAD_istruct](#) qs)
Função que obtém o top dos 10 melhores contribuidores.

4.11.1 Function Documentation

4.11.1.1 `getContributorName()`

```
char* getContributorName (
    TAD_istruct qs,
    long id )
```

Função que obtém o nome de um contribuidor através do seu ID.

Verifica se aquele contribuidor existe e, caso exista, retorna o seu campo username.

See also

[getUser\(\)](#)

Parameters

<i>qs</i>	A estrutura geral do programa.
<i>id</i>	O ID do contribuidor a pesquisar.

Returns

O username do contribuidor ou NULL.

Definition at line 50 of file users.c.

4.11.1.2 `getTop10Contributors()`

```
long* getTop10Contributors (
    TAD_istruct qs )
```

Função que obtém o top dos 10 melhores contribuidores.

Cria um iterador da hashtable correspondente aos utilizadores. Cria e inicializa um array de 10 utilizadores. Itera pela hashtable e compara o número de contribuições do utilizador a ser iterado com o top, começando pelo último lugar deste. Caso seja maior ou igual que o último, continua a fazer comparações com os acima. Caso o número de contribuições entre dois utilizadores forem iguais, é feita a decisão pelo ID destes. Caso entre no top, é feito o reajuste dos lugares abaixo.

See also

[getHashtableIterator\(\)](#)
[getNextFromIterator\(\)](#)

Parameters

<i>qs</i>	A estrutura geral do programa.
-----------	--------------------------------

Returns

Um array com os IDs dos 10 melhores contribuidores.

Definition at line 79 of file users.c.

4.11.1.3 onPageUsers()

```
TAD_istruct onPageUsers (  
    TAD_istruct qs,  
    long id,  
    char * username,  
    long articleId,  
    long revisionId )
```

Função que trata os dados relativos a users, recebidos do parsing do XML, e os envia para a hashtable.

Verifica se aquele artigo existe e se aquela revisão também já existe e, nesse caso, não modifica nada. Caso contrário, chama as funções de update da hashtable.

Parameters

<i>qs</i>	A estrutura geral do programa.
<i>id</i>	A ID do contribuidor da revisão.
<i>username</i>	O username do contribuidor da revisão.
<i>articleId</i>	O ID do artigo.
<i>revisionId</i>	O ID da revisão.

Returns

A estrutura geral atualizada.

Definition at line 23 of file users.c.

Index

all_articles
 interface.c, [33](#)
 interface.h, [37](#)
all_revisions
 interface.c, [34](#)
 interface.h, [37](#)
allArticles
 TCD_istruct, [7](#)
allRevisions
 TCD_istruct, [8](#)
article, [5](#)
 id, [5](#)
 nWords, [5](#)
 revisions, [6](#)
 size, [6](#)
 title, [6](#)
article_timestamp
 interface.c, [34](#)
 interface.h, [37](#)
article_title
 interface.c, [34](#)
 interface.h, [37](#)
articles
 TCD_istruct, [8](#)
articles.c
 get_all_articles, [12](#)
 get_all_revisions, [12](#)
 get_article_timestamp, [12](#)
 get_article_title, [13](#)
 get_unique_articles, [14](#)
 getTitlesWithPrefix, [14](#)
 getTop20LargestArticles, [15](#)
 getTopNArticlesWithMoreWords, [15](#)
 isTitlePrefix, [16](#)
 onPageArticles, [16](#)
 wordCounter, [17](#)
articles.h
 get_all_articles, [18](#)
 get_all_revisions, [18](#)
 get_article_timestamp, [19](#)
 get_article_title, [19](#)
 get_unique_articles, [20](#)
 getTitlesWithPrefix, [20](#)
 getTop20LargestArticles, [21](#)
 getTopNArticlesWithMoreWords, [22](#)
 onPageArticles, [22](#)
C:/Users/Vitor Castro/Desktop/grupo69/articles.c, [11](#)
C:/Users/Vitor Castro/Desktop/grupo69/articles.h, [17](#)
C:/Users/Vitor Castro/Desktop/grupo69/hashtable.c, [23](#)
C:/Users/Vitor Castro/Desktop/grupo69/hashtable.h, [28](#)
C:/Users/Vitor Castro/Desktop/grupo69/interface.c, [33](#)
C:/Users/Vitor Castro/Desktop/grupo69/interface.h, [36](#)
C:/Users/Vitor Castro/Desktop/grupo69/parser.c, [40](#)
C:/Users/Vitor Castro/Desktop/grupo69/parser.h, [41](#)
C:/Users/Vitor Castro/Desktop/grupo69/program.c, [42](#)
C:/Users/Vitor Castro/Desktop/grupo69/users.c, [42](#)
C:/Users/Vitor Castro/Desktop/grupo69/users.h, [44](#)
clean
 interface.c, [34](#)
 interface.h, [38](#)
clean_everything
 hashtable.c, [24](#)
 hashtable.h, [29](#)
contributions
 user, [9](#)
contributor_name
 interface.c, [34](#)
 interface.h, [38](#)
freeliterator
 hashtable.c, [24](#)
 hashtable.h, [29](#)
get_all_articles
 articles.c, [12](#)
 articles.h, [18](#)
get_all_revisions
 articles.c, [12](#)
 articles.h, [18](#)
get_article_timestamp
 articles.c, [12](#)
 articles.h, [19](#)
get_article_title
 articles.c, [13](#)
 articles.h, [19](#)
get_unique_articles
 articles.c, [14](#)
 articles.h, [20](#)
getArticle
 hashtable.c, [24](#)
 hashtable.h, [29](#)
getContributorName
 users.c, [43](#)
 users.h, [45](#)
getHashtableIterator
 hashtable.c, [25](#)
 hashtable.h, [30](#)
getNextFromIterator
 hashtable.c, [25](#)

- hashtable.h, 30
- getRevision
 - hashtable.c, 25
 - hashtable.h, 30
- getTitlesWithPrefix
 - articles.c, 14
 - articles.h, 20
- getTop10Contributors
 - users.c, 43
 - users.h, 45
- getTop20LargestArticles
 - articles.c, 15
 - articles.h, 21
- getTopNArticlesWithMoreWords
 - articles.c, 15
 - articles.h, 22
- getUser
 - hashtable.c, 25
 - hashtable.h, 31
- hashtable.c
 - clean_everything, 24
 - freeliterator, 24
 - getArticle, 24
 - getHashtableIterator, 25
 - getNextFromIterator, 25
 - getRevision, 25
 - getUser, 25
 - initHashtables, 26
 - insertOrUpdateArticle, 26
 - insertOrUpdateUser, 27
- hashtable.h
 - clean_everything, 29
 - freeliterator, 29
 - getArticle, 29
 - getHashtableIterator, 30
 - getNextFromIterator, 30
 - getRevision, 30
 - getUser, 31
 - initHashtables, 31
 - insertOrUpdateArticle, 31
 - insertOrUpdateUser, 32
 - TAD_istruct, 29
- id
 - article, 5
 - revision, 7
 - user, 9
- init
 - interface.c, 35
 - interface.h, 38
- initHashtables
 - hashtable.c, 26
 - hashtable.h, 31
- insertOrUpdateArticle
 - hashtable.c, 26
 - hashtable.h, 31
- insertOrUpdateUser
 - hashtable.c, 27
- hashtable.h, 32
- interface.c
 - all_articles, 33
 - all_revisions, 34
 - article_timestamp, 34
 - article_title, 34
 - clean, 34
 - contributor_name, 34
 - init, 35
 - load, 35
 - titles_with_prefix, 35
 - top_10_contributors, 35
 - top_20_largest_articles, 35
 - top_N_articles_with_more_words, 36
 - unique_articles, 36
- interface.h
 - all_articles, 37
 - all_revisions, 37
 - article_timestamp, 37
 - article_title, 37
 - clean, 38
 - contributor_name, 38
 - init, 38
 - load, 38
 - TAD_istruct, 37
 - titles_with_prefix, 38
 - top_10_contributors, 39
 - top_20_largest_articles, 39
 - top_N_articles_with_more_words, 39
 - unique_articles, 39
- isTitlePrefix
 - articles.c, 16
- load
 - interface.c, 35
 - interface.h, 38
- main
 - program.c, 42
- nWords
 - article, 5
- onPageArticles
 - articles.c, 16
 - articles.h, 22
- onPageUsers
 - users.c, 44
 - users.h, 46
- parseWikiData
 - parser.c, 40
 - parser.h, 41
- parser.c
 - parseWikiData, 40
- parser.h
 - parseWikiData, 41
 - TAD_istruct, 41
- program.c

- main, [42](#)
- revision, [6](#)
 - id, [7](#)
 - timestamp, [7](#)
- revisions
 - article, [6](#)
- size
 - article, [6](#)
- TAD_istruct
 - hashtable.h, [29](#)
 - interface.h, [37](#)
 - parser.h, [41](#)
- TCD_istruct, [7](#)
 - allArticles, [7](#)
 - allRevisions, [8](#)
 - articles, [8](#)
 - uniqueArticles, [8](#)
 - users, [8](#)
- timestamp
 - revision, [7](#)
- title
 - article, [6](#)
- titles_with_prefix
 - interface.c, [35](#)
 - interface.h, [38](#)
- top_10_contributors
 - interface.c, [35](#)
 - interface.h, [39](#)
- top_20_largest_articles
 - interface.c, [35](#)
 - interface.h, [39](#)
- top_N_articles_with_more_words
 - interface.c, [36](#)
 - interface.h, [39](#)
- unique_articles
 - interface.c, [36](#)
 - interface.h, [39](#)
- uniqueArticles
 - TCD_istruct, [8](#)
- user, [8](#)
 - contributions, [9](#)
 - id, [9](#)
 - username, [9](#)
- username
 - user, [9](#)
- users
 - TCD_istruct, [8](#)
- users.c
 - getContributorName, [43](#)
 - getTop10Contributors, [43](#)
 - onPageUsers, [44](#)
- users.h
 - getContributorName, [45](#)
 - getTop10Contributors, [45](#)
 - onPageUsers, [46](#)
- wordCounter
 - articles.c, [17](#)