

**ModuloRecommender**

1.0.0

Generated by Doxygen 1.5.9

Thu Oct 15 13:10:31 2009



# Contents

<b>1</b>	<b>Class Documentation</b>	<b>1</b>
1.1	Apriori Class Reference . . . . .	1
1.1.1	Detailed Description . . . . .	3
1.1.2	Constructor & Destructor Documentation . . . . .	3
1.1.2.1	Apriori . . . . .	3
1.2	Apriori_Trie Class Reference . . . . .	5
1.2.1	Detailed Description . . . . .	8
1.2.2	Member Data Documentation . . . . .	9
1.2.2.1	temp_counter_array . . . . .	9
1.3	LoadMiningOut Class Reference . . . . .	10
1.3.1	Detailed Description . . . . .	11
1.3.2	Member Function Documentation . . . . .	11
1.3.2.1	ReadFile . . . . .	11
1.3.2.2	searchProgrambyId . . . . .	12
1.3.2.3	controlDuplicate . . . . .	12
1.3.2.4	saveRecommender . . . . .	12
1.4	eit Class Reference . . . . .	13
1.4.1	Detailed Description . . . . .	15
1.4.2	Member Function Documentation . . . . .	15
1.4.2.1	getDescriptor_conteudo_genero . . . . .	15
1.4.2.2	getDescriptor_conteudo_subgenero . . . . .	16
1.4.2.3	getDescriptor_event_short_sinopse . . . . .	16

---

1.4.2.4	getEnd_time . . . . .	16
1.4.2.5	getStart_time . . . . .	16
1.4.2.6	setDescriptor_conteudo_genero . . . . .	16
1.4.2.7	setDescriptor_conteudo_subgenero . . . . .	17
1.4.2.8	setDescriptor_event_short_sinopse . . . . .	17
1.4.2.9	setEnd_time . . . . .	17
1.4.2.10	setStart_time . . . . .	17
1.5	LoadCacheDatabaseInformation Class Reference . . . . .	18
1.5.1	Detailed Description . . . . .	18
1.5.2	Member Function Documentation . . . . .	18
1.5.2.1	getInstance . . . . .	18
1.5.2.2	restoreCacheInformation . . . . .	19
1.6	LoadUserDatabaseInformation Class Reference . . . . .	20
1.6.1	Detailed Description . . . . .	21
1.6.2	Member Function Documentation . . . . .	21
1.6.2.1	getInstance . . . . .	21
1.6.2.2	getUserInformation . . . . .	22
1.6.2.3	prepareDataUser . . . . .	22
1.6.2.4	saveUserInformation . . . . .	22
1.7	sdt Class Reference . . . . .	23
1.7.1	Detailed Description . . . . .	24
1.7.2	Member Function Documentation . . . . .	25
1.7.2.1	getServiceName . . . . .	25
1.7.2.2	getServiceProviderName . . . . .	25
1.7.2.3	setServiceName . . . . .	25
1.7.2.4	setServiceProviderName . . . . .	25
1.8	si Class Reference . . . . .	26
1.8.1	Detailed Description . . . . .	28
1.9	UtilsFilterAgent Class Reference . . . . .	29
1.9.1	Detailed Description . . . . .	29
1.9.2	Member Function Documentation . . . . .	29

---

1.9.2.1	verifyDuration . . . . .	29
1.9.2.2	getValueByTime . . . . .	30
1.9.2.3	getPercentualByTime . . . . .	30
1.9.2.4	getPercentualByTime . . . . .	30
1.9.2.5	getRandom . . . . .	31
1.10	CallBack Class Reference . . . . .	32
1.10.1	Detailed Description . . . . .	33
1.10.2	Constructor & Destructor Documentation . . . . .	33
1.10.2.1	CallBack . . . . .	33
1.10.3	Member Function Documentation . . . . .	33
1.10.3.1	exe . . . . .	33
1.10.3.2	getDataCallBack . . . . .	34
1.10.3.3	getHeadCallBack . . . . .	34
1.11	Database Class Reference . . . . .	35
1.11.1	Detailed Description . . . . .	36
1.11.2	Member Function Documentation . . . . .	36
1.11.2.1	conectar . . . . .	36
1.11.2.2	insere . . . . .	36
1.11.2.3	lista . . . . .	37
1.12	LinceUtil Class Reference . . . . .	38
1.12.1	Detailed Description . . . . .	40
1.13	ListeningEvent Class Reference . . . . .	41
1.13.1	Detailed Description . . . . .	41
1.14	User Class Reference . . . . .	42
1.14.1	Detailed Description . . . . .	44
1.14.2	Member Function Documentation . . . . .	44
1.14.2.1	getCode . . . . .	44
1.14.2.2	setCode . . . . .	44
1.14.2.3	getKey . . . . .	44
1.14.2.4	setKey . . . . .	44
1.14.2.5	getDocid . . . . .	45

---

1.14.2.6	setDocid . . . . .	45
1.14.2.7	getFilename . . . . .	45
1.14.2.8	setFilename . . . . .	45
1.14.2.9	getHour . . . . .	46
1.14.2.10	setHour . . . . .	46
1.14.2.11	getMinute . . . . .	46
1.14.2.12	setMinute . . . . .	46
1.14.2.13	getSecond . . . . .	46
1.14.2.14	setSecond . . . . .	47
1.14.2.15	getDay . . . . .	47
1.14.2.16	setDay . . . . .	47
1.14.2.17	getMonth . . . . .	47
1.14.2.18	setMonth . . . . .	47
1.14.2.19	getYear . . . . .	48
1.14.2.20	setYear . . . . .	48
1.14.2.21	getWeekDay . . . . .	48
1.14.2.22	setWeekDay . . . . .	48
1.14.2.23	getTimeDuration . . . . .	48
1.14.2.24	setTimeDuration . . . . .	49
1.14.2.25	getFormatDate . . . . .	49
1.14.2.26	getFormatTime . . . . .	49
1.15	Mining Class Reference . . . . .	50
1.15.1	Detailed Description . . . . .	52
1.15.2	Member Function Documentation . . . . .	53
1.15.2.1	execute . . . . .	53
1.15.2.2	encode . . . . .	53
1.15.3	Member Data Documentation . . . . .	53
1.15.3.1	code_file_out . . . . .	53
1.16	NotifyUser Class Reference . . . . .	54
1.16.1	Detailed Description . . . . .	55
1.16.2	Member Function Documentation . . . . .	56

---

1.16.2.1	playNotify . . . . .	56
1.16.2.2	stopNotify . . . . .	56
1.16.2.3	refInstance . . . . .	56
1.17	Scheduler Class Reference . . . . .	57
1.17.1	Detailed Description . . . . .	58
1.17.2	Member Function Documentation . . . . .	58
1.17.2.1	getInstance . . . . .	58
1.17.2.2	startRecommenderModule . . . . .	58
1.17.2.3	checkRecommenderModule . . . . .	58
1.17.2.4	getThread . . . . .	59
1.17.3	Member Data Documentation . . . . .	59
1.17.3.1	_instance . . . . .	59
1.18	Edge Struct Reference . . . . .	60
1.18.1	Detailed Description . . . . .	61
1.19	Input_Output_Manager Class Reference . . . . .	62
1.19.1	Detailed Description . . . . .	64
1.19.2	Member Data Documentation . . . . .	64
1.19.2.1	new_code . . . . .	64
1.19.2.2	new_code_inverse . . . . .	65
1.20	Trie Class Reference . . . . .	66
1.20.1	Detailed Description . . . . .	68
1.20.2	Member Data Documentation . . . . .	68
1.20.2.1	edgevector . . . . .	68



# Chapter 1

## Class Documentation

### 1.1 Apriori Class Reference

```
#include <Apriori.hpp>
```



## Private Member Functions

- void **support** (const itemtype &candidate\_size)

## Private Attributes

- **Apriori\_Trie** \* **apriori\_trie**

*Abstração de uma árvore recursiva que armazena os item mais frequentes e portanto candidatos.*

- **Input\_Output\_Manager** **input\_output\_manager**

*Input\_output\_manager é responsável por gerenciar as operações de leitura e escrita.*

- map< vector< itemtype >, unsigned long > **reduced\_baskets**

*Armazenar as informações do banco de dados em memória, se o parâmetro store\_input=true;.*

- bool **store\_input**

*se store\_input = true, então as informações provinientes do arquivo de entrada serão armazenadas na memória*

### 1.1.1 Detailed Description

Essa classe implementa o algoritmo APRIORI.

O objetivo do algoritmo apriori é encontrar os conjuntos de dados frequentes L no banco de dados. Itens com pouca frequência devem ser deletados permanecendo apenas aqueles que ocorrem mais vezes.

### 1.1.2 Constructor & Destructor Documentation

#### 1.1.2.1 Apriori (ifstream & *basket\_file*, const char \* *output\_file\_name*, const bool *store\_input*)

Construtor.

#### Parameters:

*basket\_file* arquivo com o conjunto de dados para mineração

*output\_file\_name* arquivo de saída com as regras de associação

*store\_input true* indica para ler todo arquivo e armazenar as informações em memória

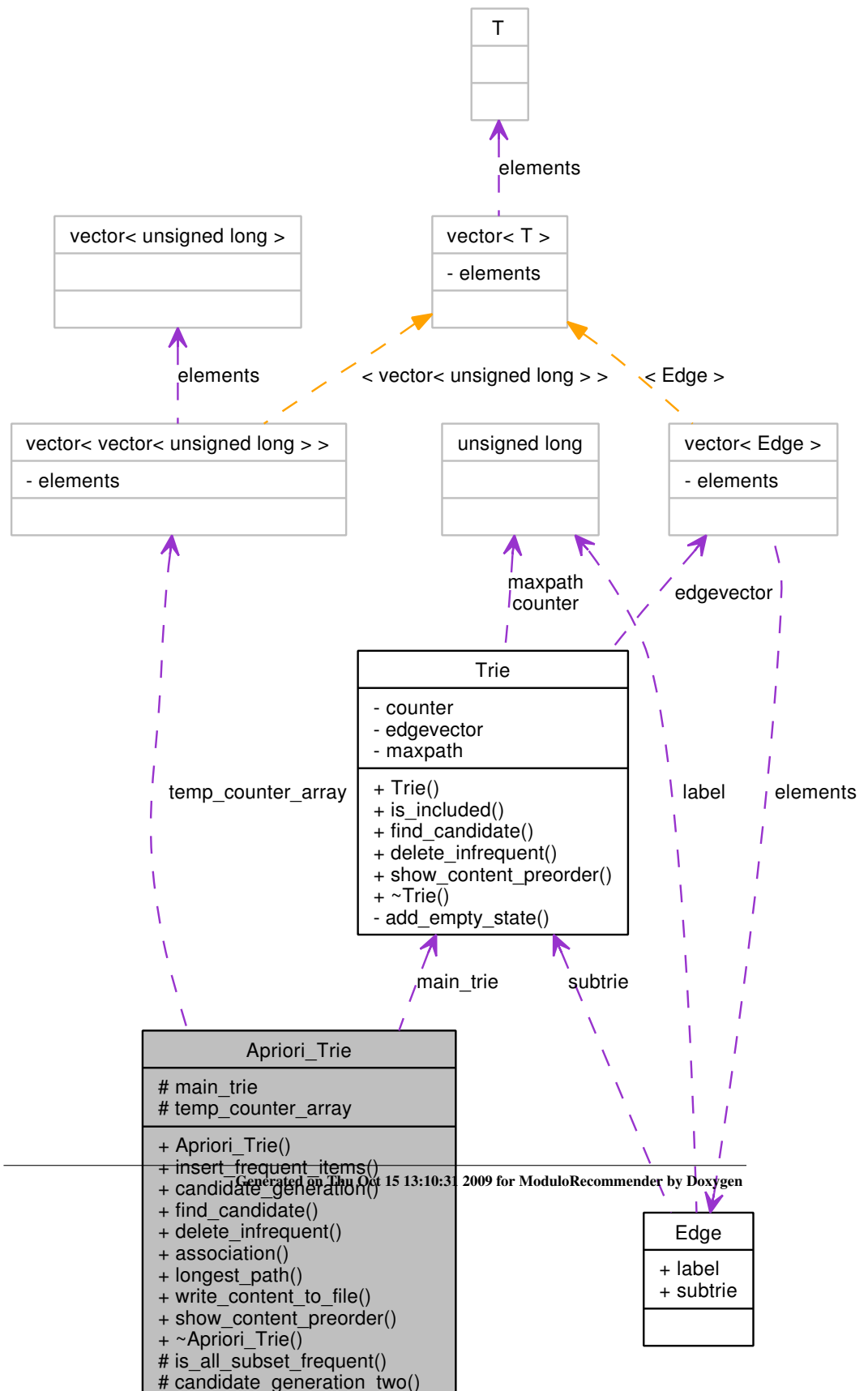
The documentation for this class was generated from the following file:

- `lince/miningagent/apriori23/Apriori.hpp`

## 1.2 Apriori\_Trie Class Reference

```
#include <Apriori_Trie.hpp>
```

Collaboration diagram for Apriori\_Trie:



## Public Member Functions

- **Apriori\_Trie** (const unsigned long counter\_of\_emptyset)
- void **insert\_frequent\_items** (const vector< unsigned long > &counters)  
*Insera os itens mais frequentes.*
- void **candidate\_generation** (const itemtype &frequent\_size)  
*Gera os candidatos a regras.*
- void **find\_candidate** (const vector< itemtype > &basket, const itemtype candidate\_size, const unsigned long counter=1)  
*Incrementa o contador dos candidatos.*
- void **delete\_infrequent** (const double min\_occurrence, const itemtype candidate\_size)  
*Deleta os itemsets menos frequentes.*
- void **association** (const double min\_conf, **Input\_Output\_Manager** &input\_output\_manager) const  
*Gera as regras de associação.*
- itemtype **longest\_path** () const  
*Retorna o tamanho do maior caminho até o nó folha da árvore (trie.hpp).*
- void **write\_content\_to\_file** (**Input\_Output\_Manager** &input\_output\_manager) const  
*Escreve os itens mais frequentes no arquivo.*
- void **show\_content\_preorder** () const  
*apresenta o conteúdo da árvore gerada em pré-ordem*

## Protected Member Functions

- bool **is\_all\_subset\_frequent** (const set< itemtype > &maybe\_candidate) const  
*Decide se todos os nós na árvore são frequentes.*
- void **candidate\_generation\_two** ()  
*Gera os candidatos com tamanho 2. Ex: Quem assiste A assiste B.*
- void **candidate\_generation\_assist** (**Trie** \***Trie**, const itemtype distance\_from\_generator, set< itemtype > &maybe\_candidate)  
*Gera os candidatos com 2 ou mais. Ex: Quem assiste A assiste B,C,D.*

- void **find\_candidate\_two** (const vector< itemtype > &basket, const unsigned long counter=1)

*Incrementa o contador dos candidatos encontrados.*

- void **delete\_infrequent\_two** (const double min\_occurrence)

*Deleta os itens menos frequentes de tamanho 2.*

- void **assoc\_rule\_find** (const double min\_conf, set< itemtype > &condition\_part, set< itemtype > &consequence\_part, const unsigned long union\_support, **Input\_Output\_Manager** &input\_output\_manager) const

*Encontra regras de associação.*

- void **assoc\_rule\_assist** (const double min\_conf, const **Trie** \***Trie**, set< itemtype > &consequence\_part, **Input\_Output\_Manager** &input\_output\_manager) const

*Geração parcial das regras de associação.*

- void **write\_content\_to\_file\_assist** (**Input\_Output\_Manager** &input\_output\_manager, const **Trie** \*actual\_state, const itemtype distance\_from\_frequent, set< itemtype > &frequent\_itemset) const

*Escreve as regras de associação encontradas no disco.*

## Protected Attributes

- **Trie main\_trie**

*Trie (p. 66) armazena os candidatos e os itens mais frequentes.*

- vector< vector< unsigned long > > **temp\_counter\_array**

### 1.2.1 Detailed Description

**Apriori\_Trie** (p. 5)

**Apriori\_Trie** (p. 5) cria a estrutura de uma árvore concebida para fornecer métodos eficientes para o algoritmo apriori.

## 1.2.2 Member Data Documentation

### 1.2.2.1 `vector< vector<unsigned long> > temp_counter_array` [protected]

`temp_counter_array` stores armazena as ocorrência de pares

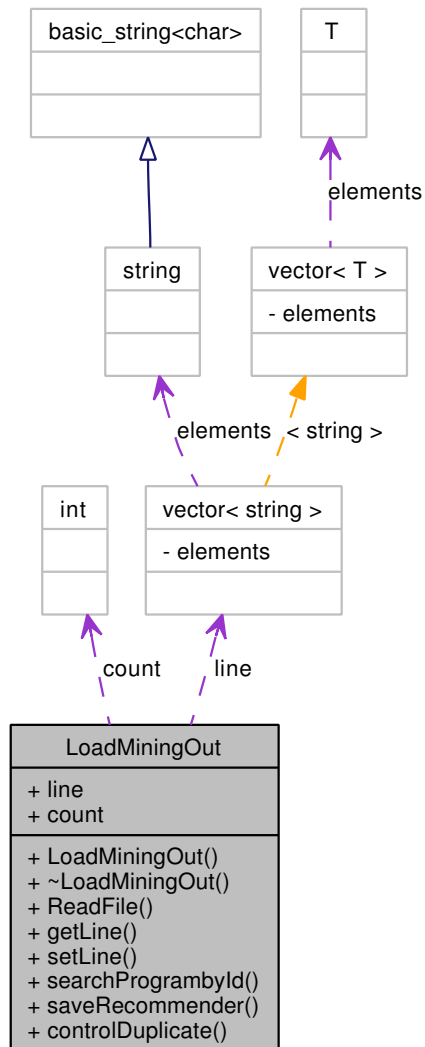
The documentation for this class was generated from the following file:

- `lince/miningagent/apriori23/Apriori_Trie.hpp`

### 1.3 LoadMiningOut Class Reference

```
#include <LoadMiningOut.h>
```

Collaboration diagram for LoadMiningOut:



#### Public Member Functions

- `int ReadFile (char *arquivo)`
- `vector< string > getLine ()`

- void **setLine** (vector< string > line)
- vector< string > **searchProgrambyId** (string genero, string subgenero, string per\_dia, vector< string > \*programacao)
- void **saveRecommender** (string arquivo, vector< string > \*user, string horario)

### Static Public Member Functions

- static void **controlDuplicate** ()

### Public Attributes

- vector< string > **line**

### Static Public Attributes

- static int **count**

## 1.3.1 Detailed Description

Responsável em persistir as informações processadas pelo módulo filterAgent em disco.

#### Author:

: Paulo Muniz de Ávila

## 1.3.2 Member Function Documentation

### 1.3.2.1 int ReadFile (char \* *arquivo*)

Leitura de arquivo texto do disco

#### Parameters:

*arquivo* nome do arquivo a ser lido

#### Returns:

número de bytes lidos do arquivo

### 1.3.2.2 `vector<string> searchProgrambyId (string genero, string subgenero, string per_dia, vector< string > * programacao)`

Busca na tabela EIT o programa recomendado

#### Parameters:

*genero* string que informa o genero do programa

*subgenero* string que informa o sub-genero do programa

*per\_dia* string que informa o dia para procurar na tabela EIT

*programacao* a tabela EIT

#### Returns:

### 1.3.2.3 `static void controlDuplicate () [static]`

Evita que ocorram duplicações nas recomendações

### 1.3.2.4 `void saveRecommender (string arquivo, vector< string > * user, string horario)`

Salva a recomendação no arquivo de formato lua. Exemplo de arquivo salvo: NOITE{  
Genero Cod = "0x0", SubGenero Cod = "0x02", Inicio = "2008-03-12 23:32:00", Fim =  
"2008-03-13 00:29:00", Programa = "REPORTER RECORD SG", Emissora = "006",  
}

#### Parameters:

*arquivo* nome do arquivo para salvar as recomendações

*user* recomendações ofertadas para o usuário

*horario* que pode ser: madrugada, manha, tarde e noite

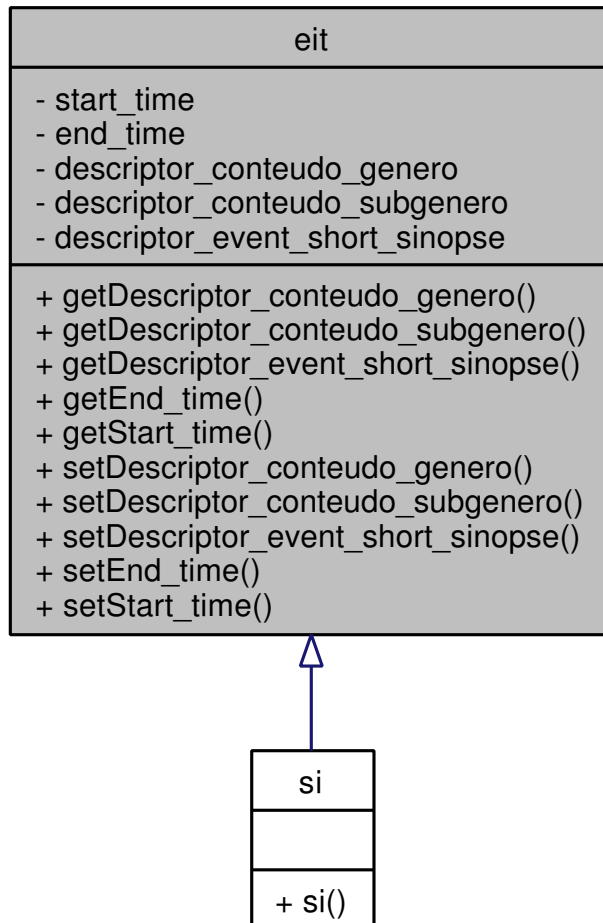
The documentation for this class was generated from the following file:

- lince/dataagent/LoadMiningOut.h

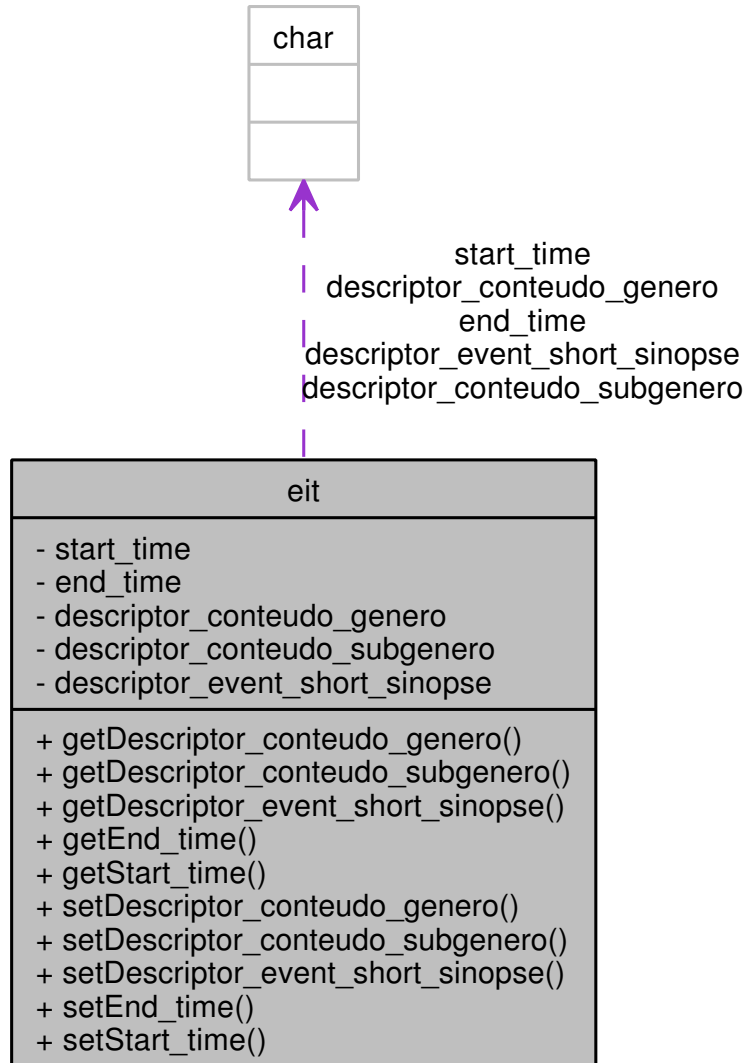
## 1.4 eit Class Reference

```
#include <eit.h>
```

Inheritance diagram for eit:



Collaboration diagram for eit:



## Public Member Functions

- char \* **getDescriptor\_conteudo\_genero** ()
- char \* **getDescriptor\_conteudo\_subgenero** ()
- char \* **getDescriptor\_event\_short\_sinopse** ()
- char \* **getEnd\_time** ()
- char \* **getStart\_time** ()

- void **setDescription\_conteudo\_genero** (const char \***descriptor\_conteudo\_genero**)
- void **setDescription\_conteudo\_subgenero** (const char \***descriptor\_conteudo\_subgenero**)
- void **setDescription\_event\_short\_sinopse** (const char \***descriptor\_event\_short\_sinopse**)
- void **setEnd\_time** (const char \***end\_time**)
- void **setStart\_time** (const char \***start\_time**)

### Private Attributes

- char **start\_time** [24]  
*inicio do programa - máxmo de 12 bytes*
- char **end\_time** [24]  
*término do programa - máximo de 12 bytes*
- char **descriptor\_conteudo\_genero** [5]  
*gênero do programa - máximo de 4 bytes*
- char **descriptor\_conteudo\_subgenero** [5]  
*sub-gênero do programa - máximo de 4 bytes*
- char **descriptor\_event\_short\_sinopse** [129]  
*sinopse do programa máximo de 128 bytes*

### 1.4.1 Detailed Description

Fornecer uma abstração para a tabela Eit. A tabela Eit é responsável em fornecer informações sobre os serviços disponibilizados pelos provedores de serviços

#### Author:

Paulo Muniz de Ávila

### 1.4.2 Member Function Documentation

#### 1.4.2.1 char\* getDescription\_conteudo\_genero ()

Recuperar a informação com o gênero do serviço Ver Norma NBR 15603-2 para maiores informações

**Returns:**

um char \* com o gênero do serviço

**1.4.2.2 char\* getDescriptor\_conteudo\_subgenero ()**

Recuperar as informações com o sub-gênero do serviço Ver Norma NBR 15603-2 para maiores informações

**Returns:**

um char \* com o sub-gênero do serviço

**1.4.2.3 char\* getDescriptor\_event\_short\_sinopse ()**

Recuperar as informações de sinopse do serviço

**Returns:**

Um ponteiro de 128 bytes com a sinopse

**1.4.2.4 char\* getEnd\_time ()**

Recuperar o horário de término do programa

**Returns:**

char \* com o horário informado pelo provedor de serviço

**1.4.2.5 char\* getStart\_time ()**

Recuperar o horário de início do programa

**Returns:**

char \* com o horário informado pelo provedor de serviço

**1.4.2.6 void setDescriptor\_conteudo\_genero (const char \*  
descriptor\_conteudo\_genero)**

Armazenar o gênero no objeto da classe Eit

**Parameters:**

*descriptor\_conteudo\_genero* char \* com o código do gênero

**1.4.2.7 void setDescription\_subgenero (const char \*  
description\_subgenero)**

Armazenar o sub-gênero no objeto da classe Eit

**Parameters:**

*description\_subgenero* char \* com o código do gênero

**1.4.2.8 void setDescription\_event\_short\_sinopse (const char \*  
description\_event\_short\_sinopse)**

Armazenar uma sinopse no objeto da classe Eit

**Parameters:**

*description\_event\_short\_sinopse* um char \* com 128 bytes descrevendo a sinopse

**1.4.2.9 void setEnd\_time (const char \* end\_time)**

Armazenar o horário de término do serviço no objeto da classe Eit

**Parameters:**

*end\_time* um char \* com o término do serviço

**1.4.2.10 void setStart\_time (const char \* start\_time)**

Armazenar o horário de início do serviço no objeto da classe Eit

**Parameters:**

*start\_time* um char \* com o horário de início do programa

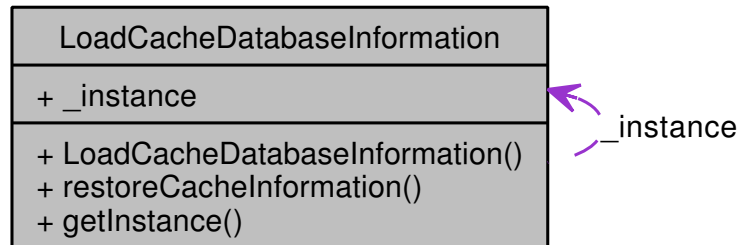
The documentation for this class was generated from the following file:

- lince/filteragent/eit.h

## 1.5 LoadCacheDatabaseInformation Class Reference

```
#include <LoadCacheDatabaseInformation.h>
```

Collaboration diagram for LoadCacheDatabaseInformation:



### Public Member Functions

- `vector< si * > * restoreCacheInformation` (string nomeCache)

### Static Public Member Functions

- `static LoadCacheDatabaseInformation * getInstance` ()

### Static Public Attributes

- `static LoadCacheDatabaseInformation * _instance`

### 1.5.1 Detailed Description

Fornecer métodos para acesso as informações contidas no cache. O cache armazena informações das tabelas SI.

#### Author:

Paulo Muniz de Ávila

### 1.5.2 Member Function Documentation

#### 1.5.2.1 `static LoadCacheDatabaseInformation* getInstance` () [static]

Retorna uma referência para um objeto da classe `LoadCacheDatabaseInformation` (p. 18)

**Returns:**

um ponteiro do tipo da classe **LoadCacheDatabaseInformation** (p. 18)

**1.5.2.2 vector<si\*>\* restoreCacheInformation (string nomeCache)**

Recupera as informações das tabelas EIT e SDT

**Parameters:**

*nomeCache* o nome do cache que contém as informações das tabelas EIT & SDT

**Returns:**

um ponteiro para um vector contendo os registros recuperados.

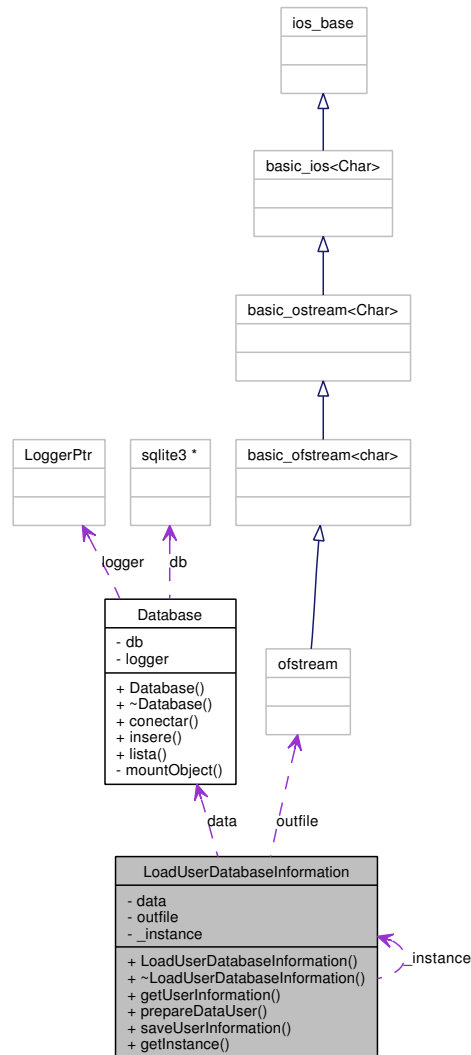
The documentation for this class was generated from the following file:

- lince/filteragent/LoadCacheDatabaseInformation.h

## 1.6 LoadUserDatabaseInformation Class Reference

```
#include <LoadUserDatabaseInformation.h>
```

Collaboration diagram for LoadUserDatabaseInformation:



### Public Member Functions

- `vector< User * > * getUserInformation ()`
- `vector< std::string > prepareDataUser (vector< User * > *user)`

- void **saveUserInformation** (vector< string > user, string file)

### Static Public Member Functions

- static **LoadUserDatabaseInformation** \* **getInstance** ()

### Private Attributes

- **Database data**  
*Objeto que representa o banco de dados sqlite.*
- ofstream **outfile**  
*classe que abstrai as operações de entrada e saída.*

### Static Private Attributes

- static **LoadUserDatabaseInformation** \* **\_instance**  
*variavel para referência ao objeto da classe **LoadUserDatabaseInformation** (p. 20)*

## 1.6.1 Detailed Description

Class LoadUserDatabaseInformation fornece métodos que permitem recuperar as informações armazenadas na base de dados sqlite e converter essas informações em objetos da classe User.

#### Author:

Paulo Muniz de Ávila

## 1.6.2 Member Function Documentation

### 1.6.2.1 static LoadUserDatabaseInformation\* getInstance () [static]

Retorna uma referência para um objeto da classe **LoadUserDatabaseInformation** (p. 20)

#### Returns:

**1.6.2.2** `vector<User*> * getUserInformation ()`

Converte informações relacionais do banco de dados em objetos da classe User.

**Returns:**

um ponteiro de vector <User\*>

**1.6.2.3** `vector<std::string> prepareDataUser (vector< User * > * user)`

Converte objetos da classe User em strings para serem submetidas aos algoritmos de mineração

**Parameters:**

*user* um ponteiro para um vector<User \*> \*user

**Returns:**

**string** com os objetos da classe User convertidos no padrão basket\_file do algoritmo apriori.

**1.6.2.4** `void saveUserInformation (vector< string > user, string file)`

Realiza a persistência das informações processadas pelo método prepareDataUser

**Parameters:**

*user* recebe uma string com os dados a serem persistidos na

*file* recebe uma string que informa o path e nome do arquivo a ser criado em disco.  
em disco.

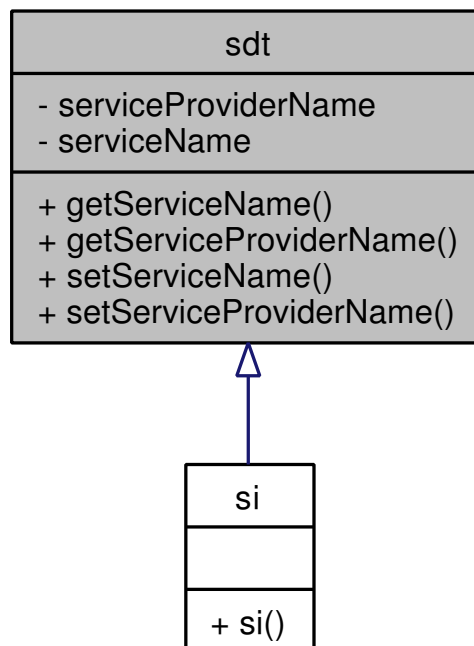
The documentation for this class was generated from the following file:

- lince/filteragent/LoadUserDatabaseInformation.h

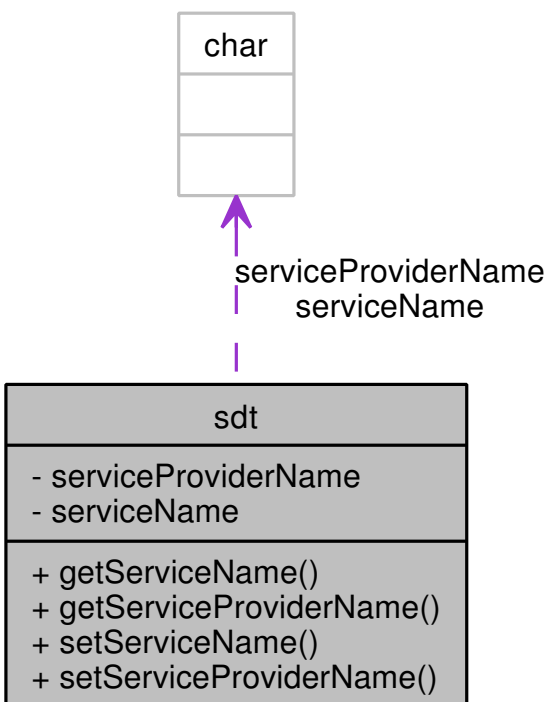
## 1.7 sdt Class Reference

```
#include <sdt.h>
```

Inheritance diagram for sdt:



Collaboration diagram for sdt:



## Public Member Functions

- char \* **getServiceName** ()
- char \* **getServiceProviderName** ()
- void **setServiceName** (const char \*serviceName)
- void **setServiceProviderName** (const char \*serviceProviderName)

## Private Attributes

- char **serviceProviderName** [128]
- char **serviceName** [128]

### 1.7.1 Detailed Description

Criar um abstração para a tabela **sdt** (p. 23) responsável em informar o nome do provedor de serviços e o serviço associado

**Author:**

Paulo Muniz de Ávila

**1.7.2 Member Function Documentation****1.7.2.1 char\* getServiceName ()**

Recuperar o nome do serviço, no sistema analógico o programa

**Returns:**

char \* com o nome do serviço

**1.7.2.2 char\* getServiceProviderName ()**

Recuperar o nome do provedor de serviços, no sistema analógico a emissora

**Returns:**

char \* com o nome do provedor de serviços

**1.7.2.3 void setName (const char \* serviceName)**

Armazenar no objeto da classe Sdt o nome do serviço

**Parameters:**

*serviceName* char \* com o nome do serviço

**1.7.2.4 void setServiceProviderName (const char \* serviceProviderName)**

Armanezar no objeto da classe Sdt o provedor de Serviço

**Parameters:**

*serviceProviderName* char \* com o nome do provedor de serviço

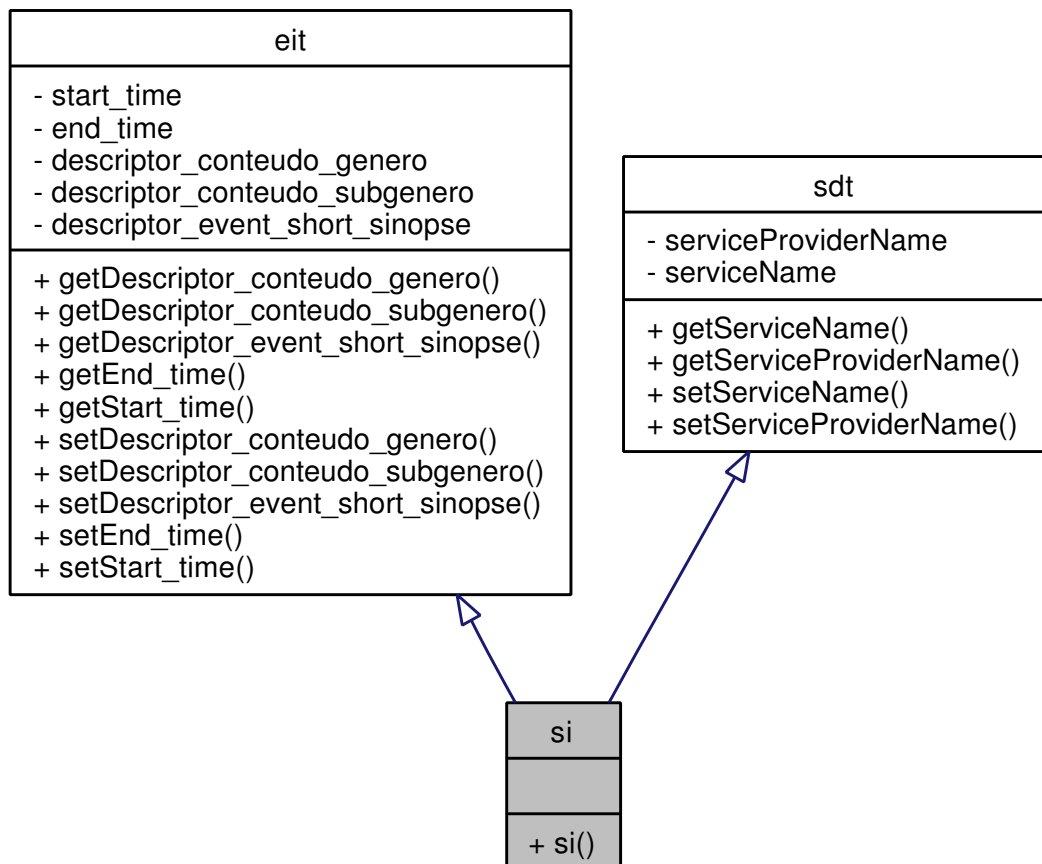
The documentation for this class was generated from the following file:

- lince/filteragent/sdt.h

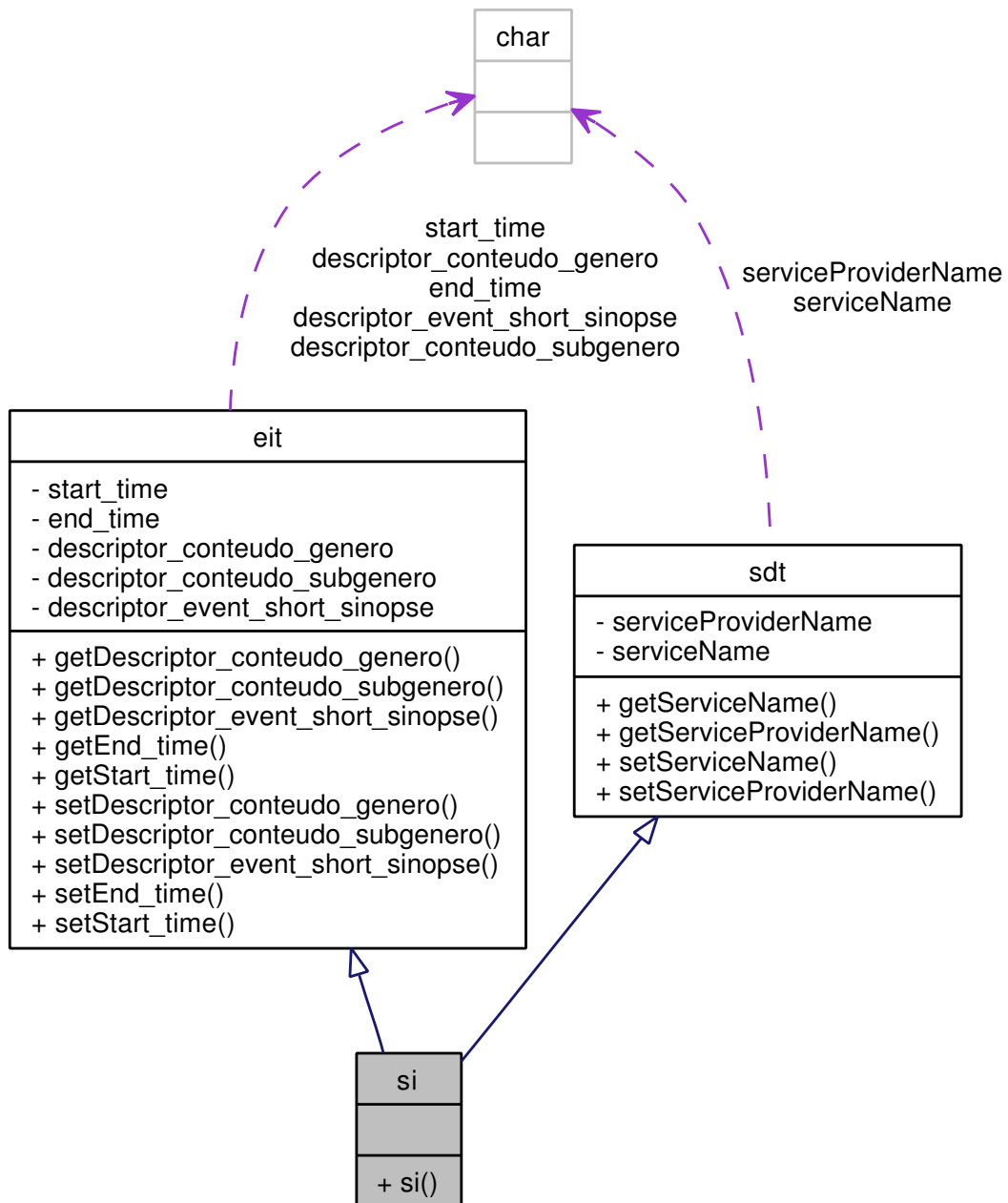
## 1.8 si Class Reference

```
#include <si.h>
```

Inheritance diagram for si:



Collaboration diagram for si:



### 1.8.1 Detailed Description

Superclasse que representa a tabela SI. As tabelas EIT & SDT fazem parte da tabela SI.

**Author:**

Paulo Muniz de Ávila

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

- `lince/filteragent/si.h`

## 1.9 UtilsFilterAgent Class Reference

```
#include <UtilsFilterAgent.h>
```

### Static Public Member Functions

- static string **getValueByTime** (string dataColectedByUser)
- static string **getPercentualByTime** (const string dataColectedByUser, vector< **si** \* > \*dataCacheByTable)
- static vector< string > **getPercentualByTime** (const string nomePrograma, double tempoUsuario, string dataInicio, vector< **si** \* > \*dataCacheByTable)
- static int **getRandom** (int min, int max)

### Static Private Member Functions

- static std::string **verifyDuration** (double duration)

#### 1.9.1 Detailed Description

Fornecer recursos para operações envolvendo os dados da cache e os dados coletados das interações do usuário

#### Author:

Paulo Muniz de Ávila

#### 1.9.2 Member Function Documentation

##### 1.9.2.1 static std::string verifyDuration (double *duration*) [static, private]

Retorna uma string representando o tempo que usuário permaneceu conectado em determinado programa.

#### Parameters:

*duration* double que representa o percentual de tempo em que o usuário ficou conectado em um determinado programa.

#### Returns:

**string** que pode ser: pouco, medio, alto

### 1.9.2.2 static string getValueByTime (string dataCollectedByUser) [static]

Calcular o período do dia em que um programa foi sintonizado. Exemplo: Programa sintonizado das 09:30hs até 11:00hs. retorna: manhã.

#### Parameters:

*dataCollectedByUser* string contendo as informações do programa, horário, data.

#### Returns:

string informando em qual horário o programa foi sintonizado

### 1.9.2.3 static string getPercentualByTime (const string dataCollectedByUser, vector< si \* > \* dataCacheByTable) [static]

Calcular o percentual de tempo que o usuário sintonizou determinando programa.

#### Parameters:

*dataCollectedByUser* string contendo as informações do programa, horário, data.

*dataCacheByTable* vector <si\*> \* um ponteiro para os dados restaurados da tabela EIT & SDT.

#### Returns:

string informando: pouco, medio,alto

### 1.9.2.4 static vector<string> getPercentualByTime (const string nomePrograma, double tempoUsuario, string dataInicio, vector< si \* > \* dataCacheByTable) [static]

Calcular o percentual de tempo que o usuário sintonizou determinando programa.

#### Parameters:

*nomePrograma* string contendo as informações do programa para comparação com as informações da EIT no cache.

*tempoUsuario* tempoUsuario double contendo o tempo em que o usuário ficou sintonizado no programa

*dataInicio* string contendo a data atual para recuperar os programas da tabela EIT & SDT

*dataCacheByTable* vector si\* um ponteiro para os dados restaurados da tabela EIT & SDT.

**Returns:**

vector contendo o tempo que o usuário assistiu cada programa.

**1.9.2.5 static int getRandom (int *min*, int *max*) [static]**

Quando não existirem dados suficiente para o algoritmo apriori processar é necessário preencher o arquivo de saída lua com informações randômicas.

**Parameters:**

*min* informa o menor valor gerado

*max* informa o máximo valor gerado

**Returns:**

**int** onde esse valor é maior igual ao minimo e menor igual ao máximo.

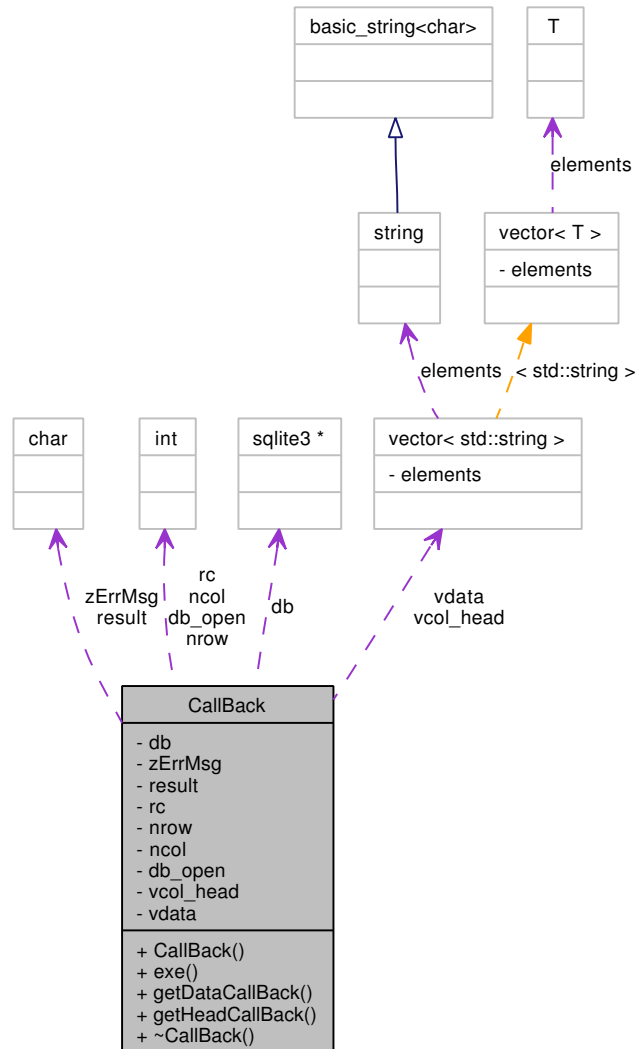
The documentation for this class was generated from the following file:

- lince/filteragent/UtilsFilterAgent.h

## 1.10 Callback Class Reference

```
#include <Callback.h>
```

Collaboration diagram for Callback:



### Public Member Functions

- **Callback** (std::string tablename)
- int **exe** (std::string s\_exe)

- `std::vector< std::string > getDataCallback ()`
- `std::vector< std::string > getHeadCallback ()`

### Private Attributes

- `sqlite3 * db`
- `char * zErrMsg`
- `char ** result`
- `int rc`
- `int nrow`
- `int ncol`
- `int db_open`
- `std::vector< std::string > vcol_head`  
*representa as colunas da tabela do banco de dados*
- `std::vector< std::string > vdata`  
*representa os dados obtidos após uma consulta sql*

#### 1.10.1 Detailed Description

Classe para recuperar informações na base de dados sqlite

**Author:**

Paulo Muniz de Ávila

#### 1.10.2 Constructor & Destructor Documentation

##### 1.10.2.1 Callback (std::string *tablename*)

Construtor Abrir o banco de dados para executar query

**Parameters:**

*tablename* nome do banco

#### 1.10.3 Member Function Documentation

##### 1.10.3.1 int exe (std::string *s\_exe*)

Executa uma query (consulta)

**Parameters:**

*s\_exe* **string** que representa a consulta a ser realizada

**Returns:**

SQLITE\_OK se tudo ocorreu com sucesso

**1.10.3.2 `std::vector<std::string> getDataCallBack ()`**

Armazena o resultado retornado a consulta ao banco de dados

**Returns:**

`vector <std::string>` com os dados recuperados da consulta sql executada

**1.10.3.3 `std::vector<std::string> getHeadCallBack ()`**

retorna os campos que compõe a tabela

**Returns:**

`vector<std::string>` com os campos que compõe a tabela

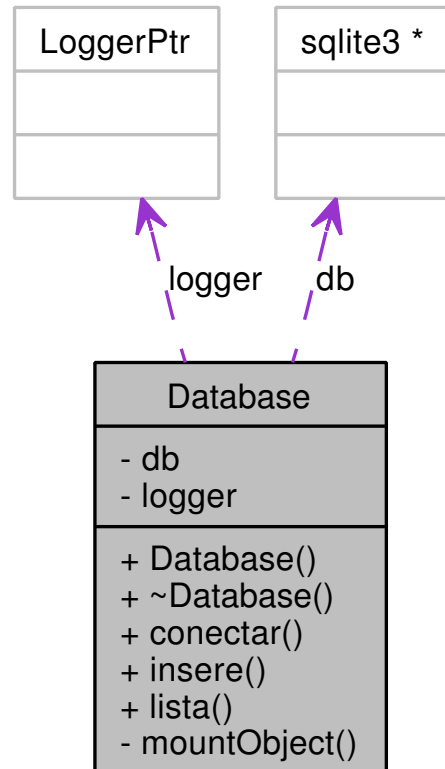
The documentation for this class was generated from the following file:

- `lince/localagent/CallBack.h`

## 1.11 Database Class Reference

```
#include <Database.h>
```

Collaboration diagram for Database:



### Public Member Functions

- bool **conectar** (std::string banco)
- int **insere** (User \*c)
- vector< User \* > \* **lista** ()

### Private Member Functions

- vector< User \* > \* **mountObject** (vector< string > retornaConsulta, vector< string > retornaHead)

*vector que recupera as informações das tabelas relacionais e criar o objetos User (p. 42)*

## Private Attributes

- `sqlite3 * db`  
*abstração para o banco de dados sqlite*
- `LoggerPtr logger`

### 1.11.1 Detailed Description

Criar ou Abrir a base de dados user.db responsável em armazenar as informações coletas pelo módulo localagent.

#### Author:

Paulo Muniz de Ávila

### 1.11.2 Member Function Documentation

#### 1.11.2.1 `bool conectar (std::string banco)`

Realizar uma conexão com o banco sqlite

#### Parameters:

*banco* nome do banco para conexão

#### Returns:

`true` se a conexão ocorrer com sucesso.

#### 1.11.2.2 `int insere (User * c)`

Insere um novo registro na base de dados user.db

#### Parameters:

*c* Objeto **User** (p. 42) com as informações data, hora, tecla

#### Returns:

`0` quando ocorre falha ou `1` para sucesso

**1.11.2.3** `vector<User*>* lista ()`

Retorna um vetor com todos registros persistidos na base de dados

**Returns:**

Um objeto da classe **User** (p. 42)

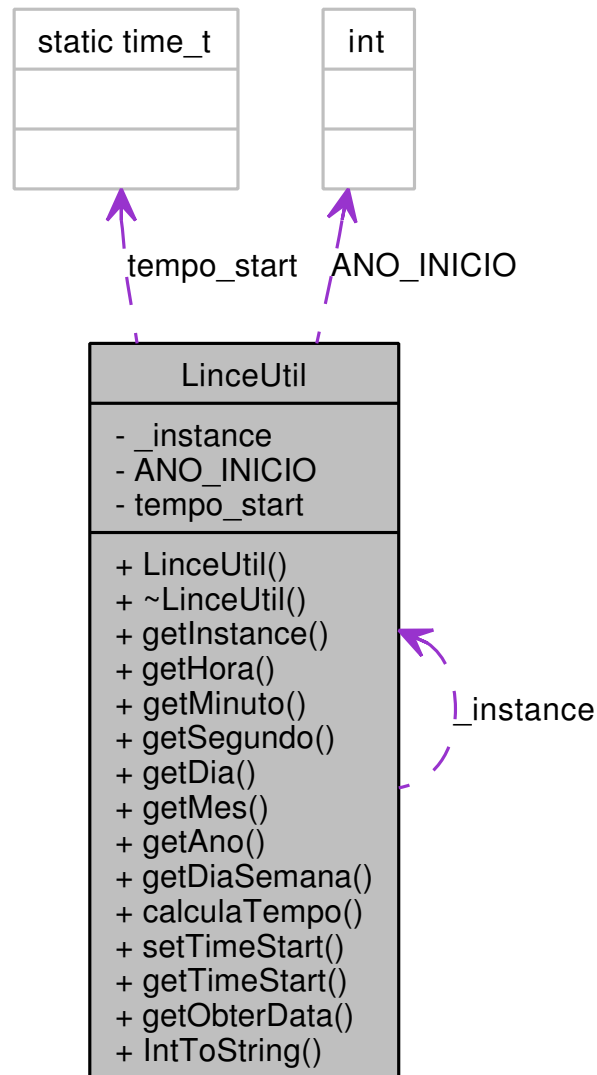
The documentation for this class was generated from the following file:

- `lince/localagent/Database.h`

## 1.12 LinceUtil Class Reference

```
#include <LinceUtil.h>
```

Collaboration diagram for LinceUtil:



### Static Public Member Functions

- static `LinceUtil * getInstance ()`

*retorna uma instância da classe para utilização dos serviços oferecidos*

- static string **getHora** ()  
*retorna a hora local*
- static string **getMinuto** ()  
*retorna os minutos da hora local*
- static string **getSegundo** ()  
*retorna os segundos da hora local*
- static string **getDia** ()  
*retorna o dia de acordo com a data atual do set-top box*
- static string **getMes** ()  
*retorna o mês de acordo com data atual do set-top box*
- static string **getAno** ()  
*retorna o ano de acordo com a data atual do set-top box*
- static string **getDiaSemana** ()  
*retorna o dia semana, sendo que domingo = 1;*
- static double **calculaTempo** (struct tm \*raw)  
*retorna o tempo que o usuário ficou em um determinado canal sem interação com o controle.*
- static void **setTimeStart** (time\_t ts)  
*define o horário em que o set-top box foi ligado*
- static time\_t **getTimeStart** ()  
*retorna o horário em que o set-top box foi ligado*
- static struct tm \* **getObterData** ()  
*retorna uma struct tm\* com a data formatada de acordo com o sistema operacional instalado no set-top box*
- static string **IntToString** (int intValue)  
*função para converter um valor do tipo inteiro em uma string*

## Static Private Attributes

- static **LinceUtil** \* **\_instance**  
*variavel do padrão singleton.*
- static int const **ANO\_INICIO** = 1900  
*define o valor inicial do ano 1900*
- static time\_t **tempo\_start**

### 1.12.1 Detailed Description

Fornece recursos para obtenção das informações coletadas a partir do monitoramento das ações do usuário

**Author:**

Paulo Muniz de Ávila

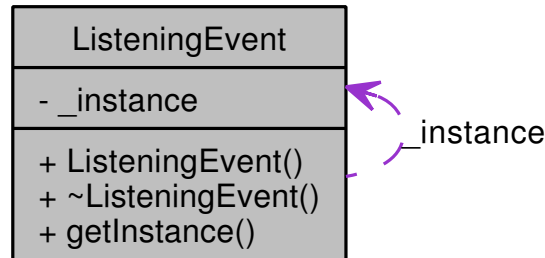
The documentation for this class was generated from the following file:

- lince/localagent/LinceUtil.h

## 1.13 ListeningEvent Class Reference

```
#include <ListeningEvent.h>
```

Collaboration diagram for ListeningEvent:



### Static Public Member Functions

- static **ListeningEvent** \* **getInstance** ()  
*retorna uma referência para a classe que permite monitor ações do usuário no set-top box*

### Static Private Attributes

- static **ListeningEvent** \* **\_instance**

#### 1.13.1 Detailed Description

Classe responsável em monitorar e coletar as ações do usuário

#### Author:

Paulo Muniz de Ávila

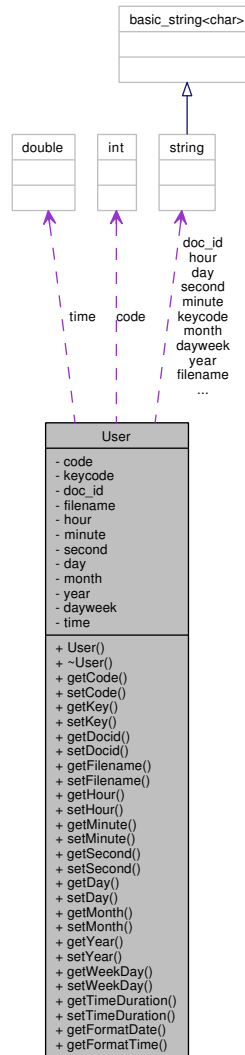
The documentation for this class was generated from the following file:

- lince/localagent/ListeningEvent.h

## 1.14 User Class Reference

```
#include <User.h>
```

Collaboration diagram for User:



### Public Member Functions

- int **getCode** ()
- void **setCode** (int setcode)

- string **getKey** ()
- void **setKey** (string setKey)
- string **getDocid** ()
- void **setDocid** (string setdocid)
- string **getFilename** ()
- void **setFilename** (string setfilename)
- string **getHour** ()
- void **setHour** (string sethour)
- string **getMinute** ()
- void **setMinute** (string setminute)
- string **getSecond** ()
- void **setSecond** (string setsecond)
- string **getDay** ()
- void **setDay** (string setday)
- string **getMonth** ()
- void **setMonth** (string setmonth)
- string **getYear** ()
- void **setYear** (string setyear)
- string **getWeekDay** ()
- void **setWeekDay** (string setweekday)
- double **getTimeDuration** ()
- void **setTimeDuration** (double settimeduration)
- string **getFormatDate** ()
- string **getFormatTime** ()

### Private Attributes

- int **code**
- string **keycode**
- string **doc\_id**
- string **filename**
- string **hour**
- string **minute**
- string **second**
- string **day**
- string **month**
- string **year**
- string **dayweek**
- double **time**

### 1.14.1 Detailed Description

Class **User** (p. 42) Representa todas as informações coletadas a partir da interação do usuário com o set-top box. Exemplo: código da tecla pressionada, nome do programa sendo assistido, hora e data.

**Author:**

Paulo Muniz de Ávila

### 1.14.2 Member Function Documentation

#### 1.14.2.1 `int getCode ()`

retorna o identificador do provedor de serviços

**Returns:**

um int que representa o provedor de serviço

#### 1.14.2.2 `void setCode (int setcode)`

Armazena o provedor de serviços sintonizado no objeto da classe user

**Parameters:**

*setcode* o código que representa o provedor de serviço

#### 1.14.2.3 `string getKey ()`

Retorna a tecla pressionada pelo usuário

**Returns:**

string informando a tecla

**See also:**

codemap

#### 1.14.2.4 `void setKey (string setKey)`

Armazena a tecla pressionada pelo usuário no objeto da classe **User** (p. 42)

**Parameters:**

*setKey* string que representa a tecla

**See also:**

codemap

**1.14.2.5 string getDocid ()**

Retorna o identificador do serviço sintonizado

**Returns:**

string que representa o serviço (canal no sistema analógico)

**1.14.2.6 void setDocid (string setdocid)**

Armazena o serviço sintonizado no objeto da classe **User** (p. 42)

**Parameters:**

*setdocid* string que identifica o serviço sintonizado

**1.14.2.7 string getFilename ()**

Retorna o nome do programa sintonizado pelo usuário

**Returns:**

string que representa o nome do programa. **Tabela EIT, descritor de conteúdo**

**1.14.2.8 void setFilename (string setfilename)**

Armazena o nome do programa sintonizado no objeto da classe **User** (p. 42)

**Parameters:**

*setfilename* nome do programa obtido da tabela EIT

**1.14.2.9 string getHour ()**

Retorna a hora atual no set-top box

**Returns:**

string informando a hora de 0 a 24.

**1.14.2.10 void setHour (string *sethour*)**

Armazena a hora atual do set-top box no objeto da classe **User** (p. 42)

**Parameters:**

*sethour* string representando a hora de 0 a 24.

**1.14.2.11 string getMinute ()**

Retorna os minutos da hora atual do set-top box

**Returns:**

string informando o minuto de 0 a 59.

**1.14.2.12 void setMinute (string *setminute*)**

Armazena os minutos atuais do set-top box no objeto da classe **User** (p. 42)

**Parameters:**

*setminute* string representando os minutos de 0 a 59.

**1.14.2.13 string getSecond ()**

Retorna os segundos da hora atual do set-top box

**Returns:**

string representando os segundos de 0 a 59.

**1.14.2.14 void setSecond (string *setsecond*)**

Armazena os segundos atuais da hora do set-top box no objeto da classe **User** (p. 42)

**Parameters:**

*setsecond* string representando os segundos de 0 a 59

**1.14.2.15 string getDay ()**

Retorna o dia de acordo com a data setada no set-top box

**Returns:**

string que representa os dias de 1 a 31.

**1.14.2.16 void setDay (string *setday*)**

Armazena o dia de acordo com a data atual do set-top Box no objeto da classe **User** (p. 42)

**Parameters:**

*setday* string que representa o dia atual de 1 a 31

**1.14.2.17 string getMonth ()**

Retorna o mês atual de acordo com a data no set-top box

**Returns:**

string que representa o mês de 1 a 12.

**1.14.2.18 void setMonth (string *setmonth*)**

Armazena o mês de acordo com a data atual do set-top Box no objeto da classe **User** (p. 42)

**Parameters:**

*setmonth* string que representa o mês de 1 a 12.

**1.14.2.19 string getYear ()**

Retorna o ano atual de acordo com a data no set-top box

**Returns:**

string que informa o ano de 1900-2027.

**1.14.2.20 void setYear (string setyear)**

Armazena o ano de acordo com a data atual do set-top Box no objeto da classe **User** (p. 42)

**Parameters:**

*setyear* string que representa o ano de 1900-2027

**1.14.2.21 string getWeekDay ()**

Retorna o dia da semana. Ex: domingo = 1, segunda = 2, ...

**Returns:**

string que representa o dia da semana 1 a 7.

**1.14.2.22 void setWeekDay (string setweekday)**

Armazena o dia da semana da data atual do set-top-box no objeto da classe **User** (p. 42)

**Parameters:**

*setweekday* string que representa o dia da semana de 1 a 7

**1.14.2.23 double getTimeDuration ()**

Retorna o tempo que um determinado serviço ficou sintonizado

**Returns:**

double que representa os segundos que um determinado serviço ficou sintonizado

**1.14.2.24 void setTimeDuration (double *settimeduration*)**

Armazena o tempo que um determinado serviço ficou sintonizado no objeto da classe **User** (p. 42)

**Parameters:**

*settimeduration* double representando o tempo em segundos

**1.14.2.25 string getFormatDate ()**

Retorna uma string com a data formatada como dd-mm-yyyy

**Returns:**

string com data preparada para inserção no banco sqlite

**1.14.2.26 string getFormatTime ()**

Retorna uma string com a hora formatada como hh:mm:ss

**Returns:**

string com a hora preparada para inserção no banco sqlite

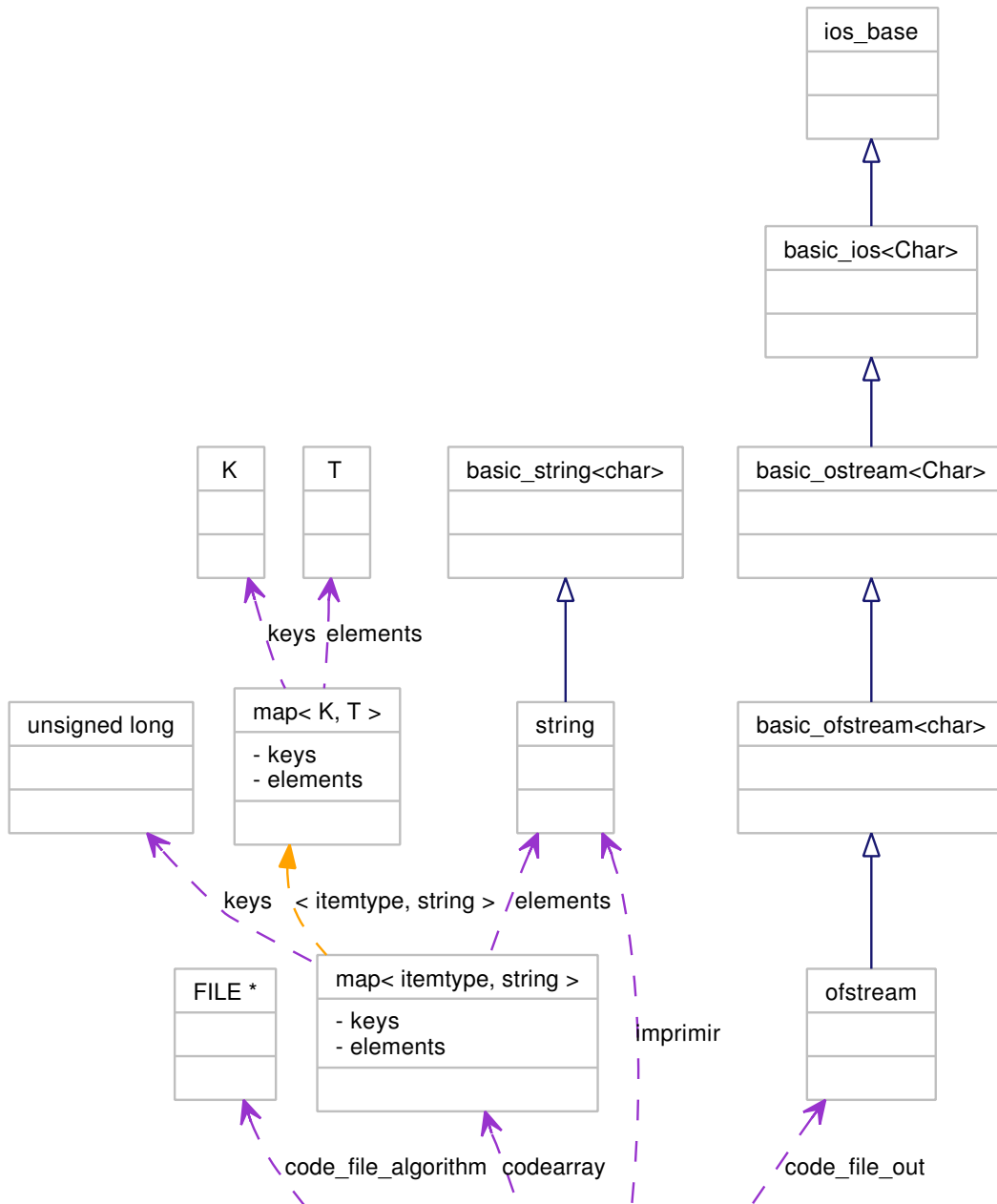
The documentation for this class was generated from the following file:

- lince/localagent/User.h

## 1.15 Mining Class Reference

```
#include <Mining.h>
```

Collaboration diagram for Mining:



Generated on Thu Oct 15 13:10:31 2009 for Module Recommender by Doxygen

```

Mining
+ code_file_algorithm
+ code_file_out
+ codearray
+ imprimir
+ Mining()
+ ~Mining()
+ execute()
+ encode()
- code_file_process()
- code_file_process_out_Algorithm()
- code_file_process_out()
- write_decoded_item()
- decode()
    
```

## Public Types

- typedef unsigned long **itemtype**

*define o tipo itemtype utilizado pelo algoritmo apriori para contagem dos elementos*

## Public Member Functions

- void **execute** (string file, double min\_sup, double min\_conf)
- void **encode** (char \*fileCodeTable, char \*fileOutMining, char \*fileOut)

## Public Attributes

- FILE \* **code\_file\_algorithm**

*abstração para o arquivo que contém os dados da interação do usuário*

- ofstream **code\_file\_out**
- map< **itemtype**, string > **codearray**

*conta a quantidade de itens (programas) de acordo com a base de dados do usuário*

- string **imprimir**

## Private Member Functions

- void **code\_file\_process** (char \*code\_file\_name)
- void **code\_file\_process\_out\_Algorithm** (char \*outAlgorithm)
- void **code\_file\_process\_out** (char \*outAlgorithm)
- void **write\_decoded\_item** (**itemtype** item)
- void **decode** ()

### 1.15.1 Detailed Description

Classe responsável em executar os serviços fornecidos pelo algoritmo **Apriori** (p. 1)

#### Author:

Paulo Muniz de Ávila

## 1.15.2 Member Function Documentation

### 1.15.2.1 void execute (string *file*, double *min\_sup*, double *min\_conf*)

Executa o algoritmo apriori

**Parameters:**

*file* arquivo que contém as interações do usuário (comportamento do usuário)

*min\_sup* parametros do algoritmo. Utilizado para desconsiderar poucas interações do usuário.

*min\_conf* Confiabilidade dos dados. Quanto maior esse parâmetro, menos regras são produzidas.

### 1.15.2.2 void encode (char \* *fileCodeTable*, char \* *fileOutMining*, char \* *fileOut*)

O algoritmo apriori produz a saída toda em formato numérico. Esse método mapea a saída do algoritmo numérico para strings.

**Parameters:**

*fileCodeTable* tabela usada para o mapeamento string -> numérico

*fileOutMining* arquivo gerado pelo algoritmo apriori (numérico)

*fileOut* arquivo de saída mapeado numérico -> string

## 1.15.3 Member Data Documentation

### 1.15.3.1 ofstream code\_file\_out

abstração para o arquivo de saída produzido pelo algoritmo apriori. O arquivo de saída ainda não está pronto, necessitando ser submetido aos algoritmos do módulo Filter-Agent

**See also:**

br.ufscar.lince.ginga.recommender.filteragent

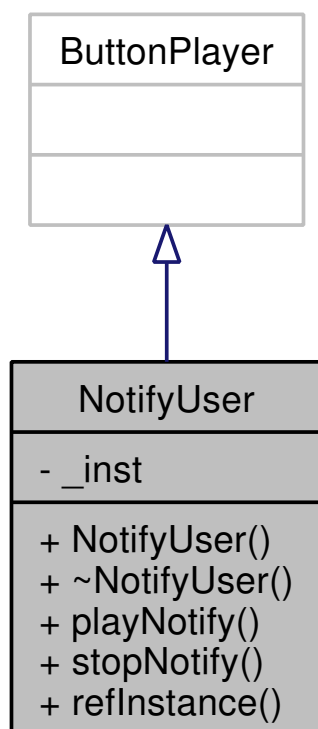
The documentation for this class was generated from the following file:

- lince/miningagent/Mining.h

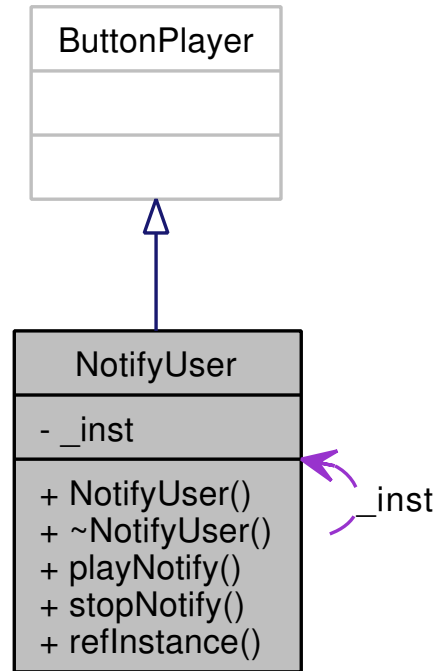
## 1.16 NotifyUser Class Reference

```
#include <NotifyUser.h>
```

Inheritance diagram for NotifyUser:



Collaboration diagram for NotifyUser:



### Public Member Functions

- void **playNotify** (string img)
- void **stopNotify** ()

### Static Public Member Functions

- static **NotifyUser** \* **refInstance** ()

### Static Private Attributes

- static **NotifyUser** \* **\_inst**

#### 1.16.1 Detailed Description

Emitir aviso visual que o processo de mineração está em andamento

**Author:**

Paulo Muniz de Ávila

**1.16.2 Member Function Documentation****1.16.2.1 void playNotify (string *img*)**

Exibe uma imagem informando que o processo de mineração de dados está ocorrendo.

**Parameters:**

*img* caminho e nome da imagem a ser exibida

**1.16.2.2 void stopNotify ()**

Ao término da mineração, remove a notificação

**1.16.2.3 static NotifyUser\* refInstance () [static]**

Singleton - Retorna uma referência da classe

**Returns:**

Um objeto do tipo **NotifyUser** (p. 54)

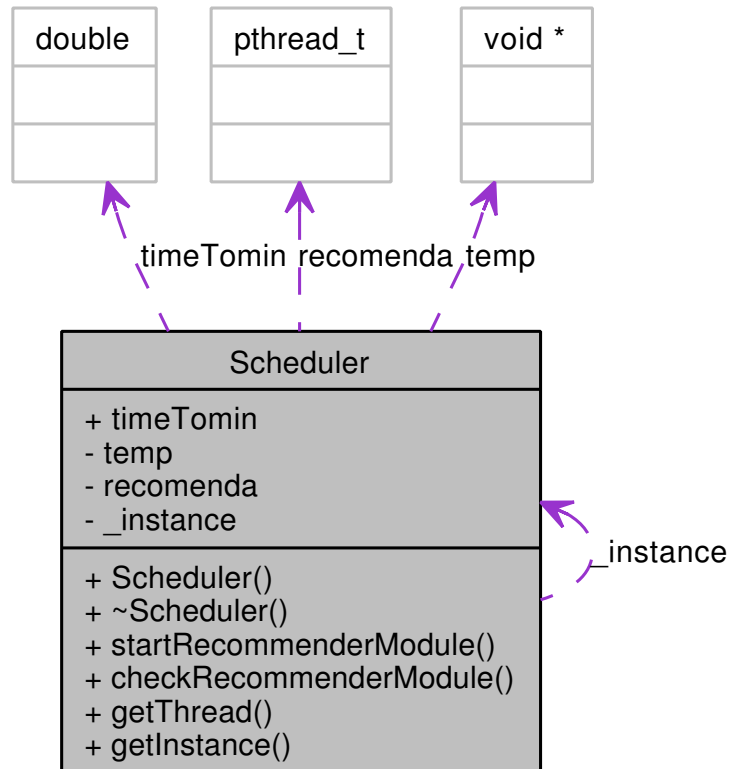
The documentation for this class was generated from the following file:

- lince/scheduleragent/NotifyUser.h

## 1.17 Scheduler Class Reference

```
#include <Scheduler.h>
```

Collaboration diagram for Scheduler:



### Public Member Functions

- void **startRecommenderModule** ()
- void **checkRecommenderModule** ()
- pthread\_t **getThread** ()

### Static Public Member Functions

- static Scheduler \* **getInstance** ()

## Public Attributes

- double **timeTomin**

## Private Attributes

- void \* **temp**
- pthread\_t **recomenda**

## Static Private Attributes

- static Scheduler \* **\_instance**

### 1.17.1 Detailed Description

Classe Responsável em disparar o processo de mineração

#### Author:

Paulo Muniz de Ávila

### 1.17.2 Member Function Documentation

#### 1.17.2.1 static Scheduler\* getInstance () [static]

Retorna uma instância da classe Agendadora

#### Returns:

Um objeto do tipo da classe **Scheduler** (p. 57)

#### 1.17.2.2 void startRecommenderModule ()

Dispara o processo responsável realizar a mineração e produzir as recomendações uma vez ao dia.

#### 1.17.2.3 void checkRecommenderModule ()

Verifica e detecta o momento de realizar a mineração dos dados

#### 1.17.2.4 pthread\_t getThread ()

Obtém acesso a thread que executa o scheduler do módulo Recommender

**Returns:**

thread

### 1.17.3 Member Data Documentation

#### 1.17.3.1 Scheduler\*\_instance [static, private]

singleton

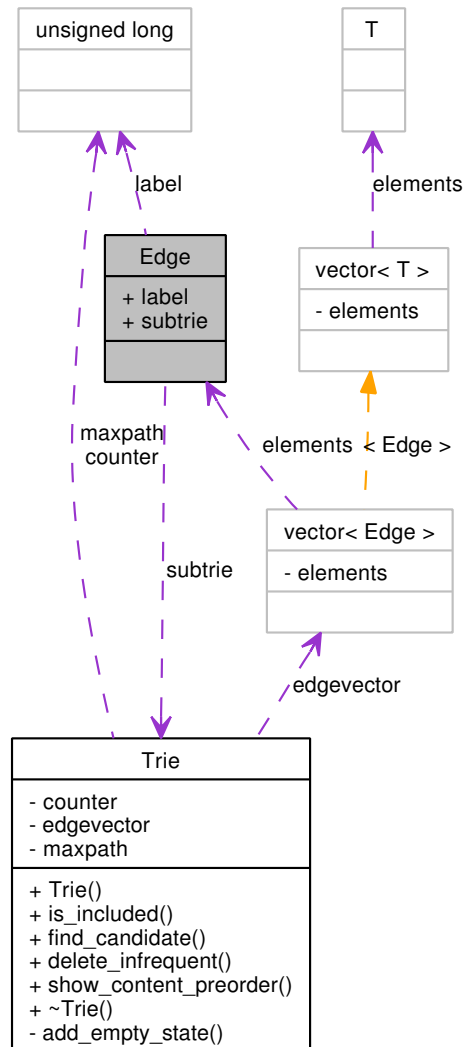
The documentation for this class was generated from the following file:

- lince/scheduleragent/Scheduler.h

## 1.18 Edge Struct Reference

```
#include <Trie.hpp>
```

Collaboration diagram for Edge:



### Public Attributes

- itemtype **label**
- **Trie** \* **subtrie**

**1.18.1 Detailed Description**

Struct que representa um **Trie** (p. 66)

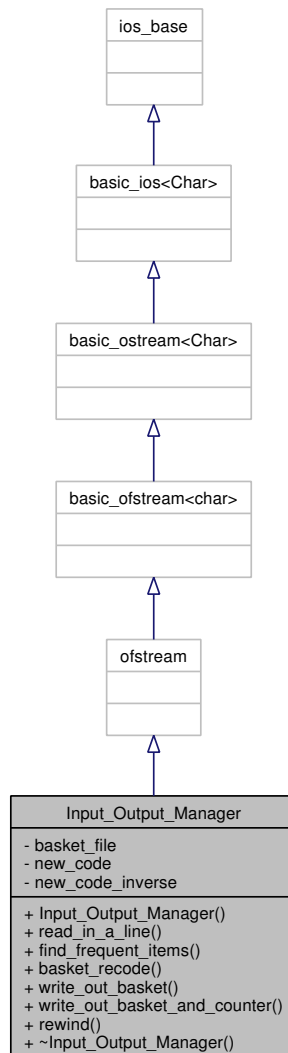
The documentation for this struct was generated from the following file:

- [lince/miningagent/apriori23/Trie.hpp](#)

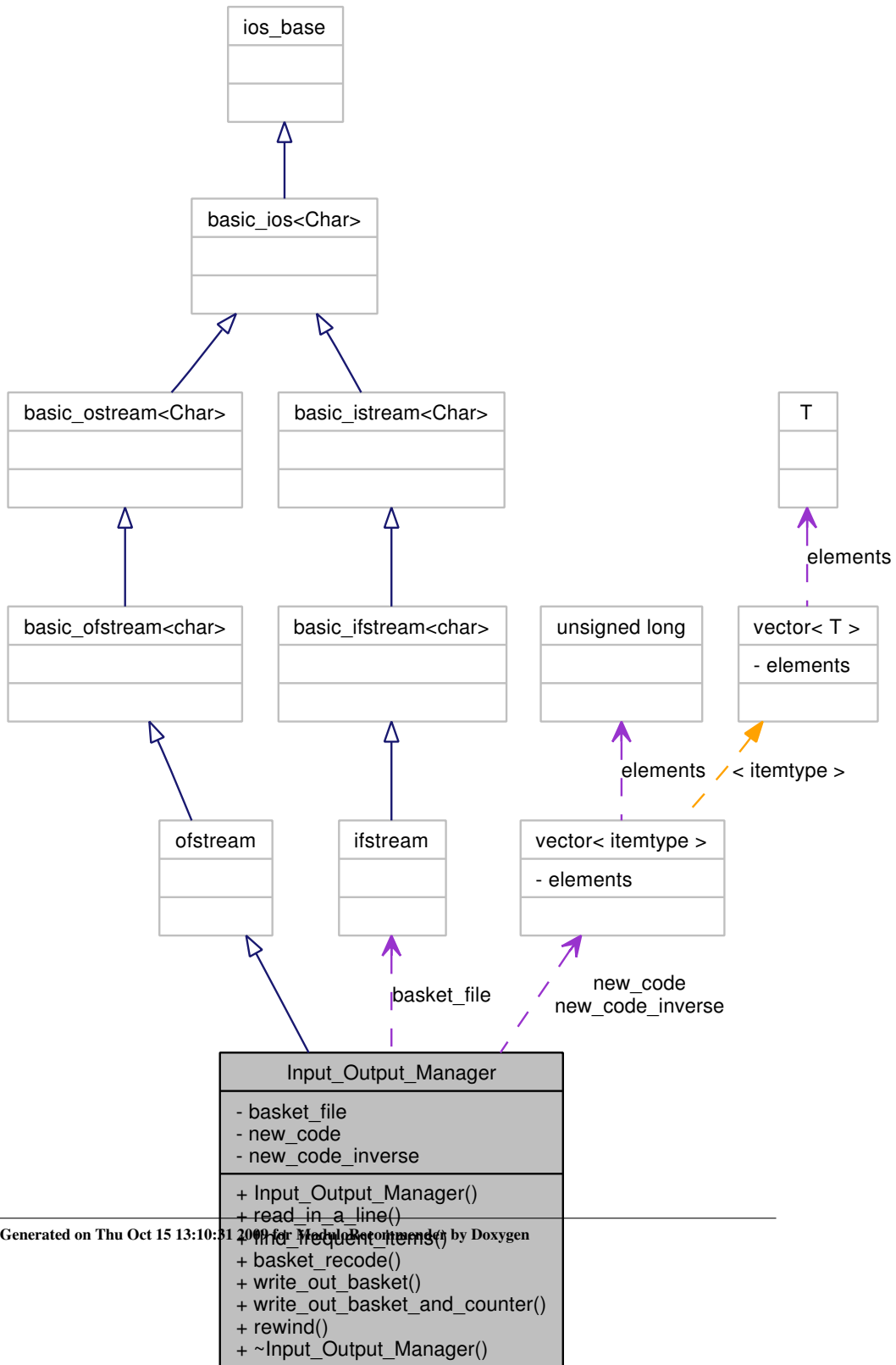
## 1.19 Input\_Output\_Manager Class Reference

```
#include <Input_Output_Manager.hpp>
```

Inheritance diagram for Input\_Output\_Manager:



Collaboration diagram for Input\_Output\_Manager:



## Public Member Functions

- **Input\_Output\_Manager** (ifstream &basket\_file, const char \*output\_file\_name)
- int **read\_in\_a\_line** (set< itemtype > &basket)  
*Realizar a leitura de uma linha do arquivo de entrada (uma transação).*
- unsigned long **find\_frequent\_items** (const double min\_supp, vector< unsigned long > &support\_of\_items)  
*Determina a frequencia de cada item.*
- void **basket\_recode** (const set< itemtype > &original\_basket, vector< itemtype > &new\_basket)  
*Criar um arquivo temporário com os itemsets mais frequentes.*
- void **write\_out\_basket** (const set< itemtype > &itemset)  
*Escreve um itemset para um arquivo.*
- void **write\_out\_basket\_and\_counter** (const set< itemtype > &itemset, const unsigned long counter)  
*Escreve o itemset e seu contador de frequencia para um arquivo.*
- void **rewind** ()

## Private Attributes

- ifstream & **basket\_file**  
*Arquivo que contém as transações. Arquivo de entrada do algoritmo.*
- vector< itemtype > **new\_code**
- vector< itemtype > **new\_code\_inverse**

### 1.19.1 Detailed Description

Essa classe é responsável pelas operações de entrada e saída (I/O Operations)

### 1.19.2 Member Data Documentation

#### 1.19.2.1 vector<itemtype> new\_code [private]

Vector que representa a frequência de cada item.

**1.19.2.2** `vector<itemtype> new_code_inverse` [private]

O inverso de new\_code vector.

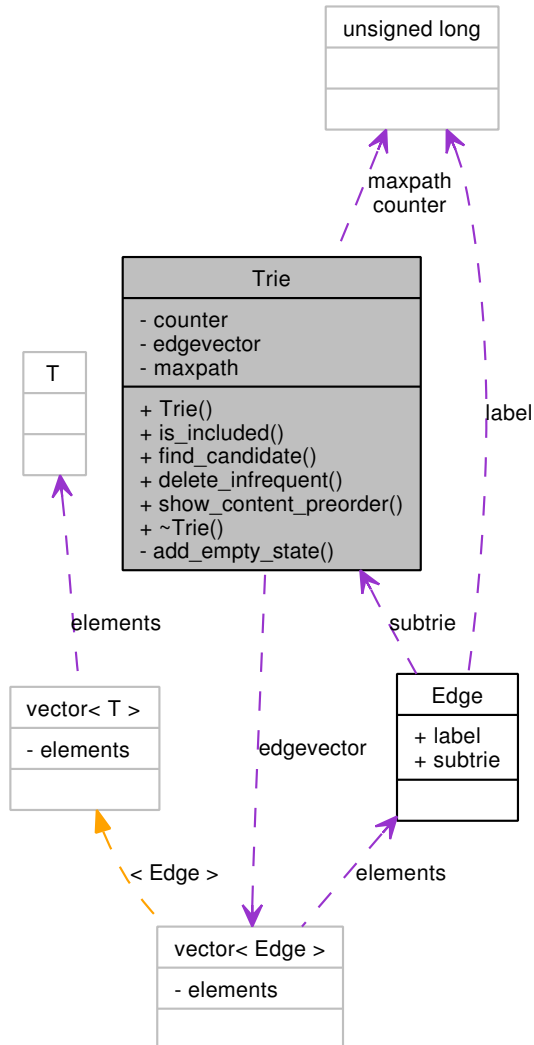
The documentation for this class was generated from the following file:

- `lince/miningagent/apriori23/Input_Output_Manager.hpp`

## 1.20 Trie Class Reference

```
#include <Trie.hpp>
```

Collaboration diagram for Trie:



### Public Member Functions

- **Trie** (const unsigned long init\_counter)

- const **Trie** \* **is\_included** (const set< itemtype > &an\_itemset, set< itemtype >::const\_iterator item\_it) const

*Decide se o itemset será ou não incluído na árvore.*

- void **find\_candidate** (vector< itemtype >::const\_iterator it\_basket\_upper\_bound, const itemtype distance\_from\_candidate, vector< itemtype >::const\_iterator it\_basket, const unsigned long counter\_incr=1)

*Incrementa o contador dos itemsets.*

- void **delete\_infrequent** (const double min\_occurrence, const itemtype distance\_from\_candidate)

*Apaga as arvores que contém os itens que ocorrem poucas vezes.*

- void **show\_content\_preorder** () const

*exibe a árvore em pré-ordem*

## Private Member Functions

- void **add\_empty\_state** (const itemtype item, const unsigned long init\_counter=0)

*adiciona um nó vazio na árvore*

## Private Attributes

- unsigned long **counter**

*contador que armazena o número de nós na trie (Árvore)*

- vector< **Edge** > **edgevector**

- itemtype **maxpath**

*armazena o tamanho do maior caminho até os nós folha da árvore (trie) inicia no nó raiz*

## Friends

- class **Apriori\_Trie**

### 1.20.1 Detailed Description

A classe trie é uma abstração para uma estrutura de dados recursivas. Cada nó raíz representa um itemset.

### 1.20.2 Member Data Documentation

#### 1.20.2.1 `vector<Edge> edgevector` [private]

`edgevector` armazena os nós raízes das trie

`edgevector` é organizado. Em linhas gerais é um vector de tries que são estruturas de dados do tipo árvores

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

- `lince/miningagent/apriori23/Trie.hpp`

# Index

`_instance`  
    `br::ufscar::lince::ginga::recommender::scheduler::Schedule`, 21  
    59  
Apriori, 1  
    Apriori, 3  
Apriori\_Trie, 5  
    `temp_counter_array`, 9  
`br::ufscar::lince::ginga::recommender::dataagent::LocalMainOrder`, 25  
    10  
    `controlDuplicate`, 12  
    `ReadFile`, 11  
    `saveRecommender`, 12  
    `searchProgrambyId`, 11  
`br::ufscar::lince::ginga::recommender::filteragent::LocalMainOrder`, 26  
    13  
    `getDescriptor_contenido_genero`, 15  
    `getDescriptor_contenido_subgenero`, 16  
    `getDescriptor_event_short_sinopse`, 16  
    `getEnd_time`, 16  
    `getStart_time`, 16  
    `setDescriptor_contenido_genero`, 16  
    `setDescriptor_contenido_subgenero`, 16  
    `setDescriptor_event_short_sinopse`, 17  
    `setEnd_time`, 17  
    `setStart_time`, 17  
`br::ufscar::lince::ginga::recommender::filteragent::LocalMainOrder`, 41  
    18  
    `getInstance`, 18  
    `restoreCacheInformation`, 19  
`br::ufscar::lince::ginga::recommender::filteragent::LocalMainOrder`, 47  
    20  
    `getInstance`, 21  
    `getCacheInformation`, 21  
    `prepareDataUser`, 22  
    `saveUserInformation`, 22  
    `br::ufscar::lince::ginga::recommender::filteragent::sdt`, 23  
    `getServiceName`, 25  
    `getServiceProviderName`, 25  
    `setServiceName`, 25  
    `getServiceName`, 25  
    `br::ufscar::lince::ginga::recommender::filteragent::si`, 26  
    `br::ufscar::lince::ginga::recommender::filteragent::UtilsFilterAgent`, 29  
    `getPercentualByTime`, 30  
    `getRandom`, 31  
    `getValueByTime`, 29  
    `verifyDuration`, 29  
    `br::ufscar::lince::ginga::recommender::localagent::CallBack`, 32  
    `CallBack`, 33  
    `exe`, 33  
    `getDataCallBack`, 34  
    `getHeadCallBack`, 34  
    `br::ufscar::lince::ginga::recommender::localagent::Database`, 35  
    `conectar`, 36  
    `insere`, 36  
    `lista`, 36  
    `br::ufscar::lince::ginga::recommender::localagent::LinceUtil`, 38  
    `br::ufscar::lince::ginga::recommender::localagent::ListeningEvent`, 41  
    `br::ufscar::lince::ginga::recommender::localagent::User`, 42  
    `br::ufscar::lince::ginga::recommender::localagent::DatabaseInformation`, 44  
    `getCache`, 44  
    `getDay`, 47

- getDocid, 45
- getFilename, 45
- getFormatDate, 49
- getFormatTime, 49
- getHour, 45
- getKey, 44
- getMinute, 46
- getMonth, 47
- getSecond, 46
- getTimeDuration, 48
- getWeekDay, 48
- getYear, 47
- setCode, 44
- setDay, 47
- setDocid, 45
- setFilename, 45
- setHour, 46
- setKey, 44
- setMinute, 46
- setMonth, 47
- setSecond, 46
- setTimeDuration, 48
- setWeekDay, 48
- setYear, 48
- br::ufscar::lince::ginga::recommender::miningagent::Mining, 50
- code\_file\_out, 53
- encode, 53
- execute, 53
- br::ufscar::lince::ginga::recommender::scheduler::NotifyUser, 54
- playNotify, 56
- refInstance, 56
- stopNotify, 56
- br::ufscar::lince::ginga::recommender::scheduler::Scheduler, 57
- \_instance, 59
- checkRecommenderModule, 58
- getInstance, 58
- getThread, 58
- startRecommenderModule, 58
- CallBack
  - br::ufscar::lince::ginga::recommender::localagent::CallBack, 33
- checkRecommenderModule
  - br::ufscar::lince::ginga::recommender::scheduler::Scheduler, 58
- code\_file\_out
  - br::ufscar::lince::ginga::recommender::miningagent::Mining, 53
- conectar
  - br::ufscar::lince::ginga::recommender::localagent::Database, 36
- controlDuplicate
  - br::ufscar::lince::ginga::recommender::dataagent::LoadMiningC, 12
- Edge, 60
- edgevector
  - Trie, 68
- encode
  - br::ufscar::lince::ginga::recommender::miningagent::Mining, 53
- exe
  - br::ufscar::lince::ginga::recommender::localagent::CallBack, 33
- execute
  - br::ufscar::lince::ginga::recommender::miningagent::Mining, 53
- getCode
  - br::ufscar::lince::ginga::recommender::localagent::User, 44
- getDataCallBack
  - br::ufscar::lince::ginga::recommender::localagent::CallBack, 34
- getDay
  - br::ufscar::lince::ginga::recommender::localagent::User, 47
- getDescriptor\_contenido\_genero
  - br::ufscar::lince::ginga::recommender::filteragent::eit, 15
- getDescriptor\_contenido\_subgenero
  - br::ufscar::lince::ginga::recommender::filteragent::eit, 16
- getDescriptor\_event\_short\_sinopse
  - br::ufscar::lince::ginga::recommender::filteragent::eit, 16
- getDocid
  - br::ufscar::lince::ginga::recommender::localagent::User, 45

getEnd_time	getServiceProviderName
br::ufscar::lince::ginga::recommender::filteragent::sdt,	br::ufscar::lince::ginga::recommender::filteragent::sdt,
16	25
getFilename	getStart_time
br::ufscar::lince::ginga::recommender::localagent::User,	br::ufscar::lince::ginga::recommender::filteragent::eit,
45	16
getFormatDate	getThread
br::ufscar::lince::ginga::recommender::localagent::User,	br::ufscar::lince::ginga::recommender::scheduler::Scheduler,
49	58
getFormatTime	getTimeDuration
br::ufscar::lince::ginga::recommender::localagent::User,	br::ufscar::lince::ginga::recommender::localagent::User,
49	48
getHeadCallBack	getUserInformation
br::ufscar::lince::ginga::recommender::localagent::CallBack;	br::ufscar::lince::ginga::recommender::filteragent::LoadUserDatabaseInforma
34	21
getHour	getValueByTime
br::ufscar::lince::ginga::recommender::localagent::User,	br::ufscar::lince::ginga::recommender::filteragent::UtilsFilterAgent,
45	29
getInstance	getWeekDay
br::ufscar::lince::ginga::recommender::filteragent::LoadCacheDatabaseInformation,	br::ufscar::lince::ginga::recommender::localagent::User,
18	48
br::ufscar::lince::ginga::recommender::filteragent::LoadUserDatabaseInformation,	br::ufscar::lince::ginga::recommender::localagent::User,
21	45
br::ufscar::lince::ginga::recommender::scheduler::Scheduler,	
58	
getKey	Input_Output_Manager, 62
br::ufscar::lince::ginga::recommender::localagent::User,	new_code, 64
44	new_code_inverse, 64
getMinute	insere
br::ufscar::lince::ginga::recommender::localagent::User,	br::ufscar::lince::ginga::recommender::localagent::Database,
46	36
getMonth	lista
br::ufscar::lince::ginga::recommender::localagent::User,	br::ufscar::lince::ginga::recommender::localagent::Database,
47	36
getPercentualByTime	br::ufscar::lince::ginga::recommender::filteragent::UtilsFilterAgent,
br::ufscar::lince::ginga::recommender::filteragent::UtilsFilterAgent,	new_code
30	Input_Output_Manager, 64
getRandom	new_code_inverse
br::ufscar::lince::ginga::recommender::filteragent::UtilsFilterAgent,	Input_Output_Manager, 64
31	
getSecond	playNotify
br::ufscar::lince::ginga::recommender::localagent::User,	br::ufscar::lince::ginga::recommender::scheduler::NotifyUser,
46	56
getServiceName	prepareDataUser
br::ufscar::lince::ginga::recommender::filteragent::sdt;	br::ufscar::lince::ginga::recommender::filteragent::LoadUserDatabaseInforma
25	22

ReadFile	setKey
br::ufscar::lince::ginga::recommender::dataagent::LoadMiningData, 11	br::ufscar::lince::ginga::recommender::localagent::User, 44
refInstance	setMinute
br::ufscar::lince::ginga::recommender::scheduler::NotifyUser, 56	br::ufscar::lince::ginga::recommender::localagent::User, 46
restoreCacheInformation	setMonth
br::ufscar::lince::ginga::recommender::filteragent::LoadCacheDatabaseInformation, 19	br::ufscar::lince::ginga::recommender::localagent::User, 47
saveRecommender	setSecond
br::ufscar::lince::ginga::recommender::dataagent::LoadMiningOut, 12	br::ufscar::lince::ginga::recommender::localagent::User, 46
saveUserInformation	setServiceName
br::ufscar::lince::ginga::recommender::filteragent::LoadUserDatabaseInformation, 22	br::ufscar::lince::ginga::recommender::filteragent::sdt, 25
searchProgrambyId	setServiceProviderName
br::ufscar::lince::ginga::recommender::dataagent::LoadMiningOut, 11	br::ufscar::lince::ginga::recommender::filteragent::sdt, 25
setCode	setStart_time
br::ufscar::lince::ginga::recommender::localagent::User, 44	br::ufscar::lince::ginga::recommender::filteragent::eit, 43
setDay	setTimeDuration
br::ufscar::lince::ginga::recommender::localagent::User, 47	br::ufscar::lince::ginga::recommender::localagent::User, 43
setDescriptor_contenido_genero	setWeekDay
br::ufscar::lince::ginga::recommender::filteragent::it, 16	br::ufscar::lince::ginga::recommender::localagent::User, 48
setDescriptor_contenido_subgenero	setYear
br::ufscar::lince::ginga::recommender::filteragent::it, 16	br::ufscar::lince::ginga::recommender::localagent::User, 48
setDescriptor_event_short_sinopse	startRecommenderModule
br::ufscar::lince::ginga::recommender::filteragent::it, 17	br::ufscar::lince::ginga::recommender::scheduler::Scheduler, 58
setDocid	stopNotify
br::ufscar::lince::ginga::recommender::localagent::User, 45	br::ufscar::lince::ginga::recommender::scheduler::NotifyUser, 56
setEnd_time	temp_counter_array
br::ufscar::lince::ginga::recommender::filteragent::it, 17	Apriori, Trie, 9
setFilename	Trie, 66
br::ufscar::lince::ginga::recommender::localagent::User, 45	edgevector, 68
setHour	verifyDuration
br::ufscar::lince::ginga::recommender::localagent::User, 46	br::ufscar::lince::ginga::recommender::filteragent::UtilsFilterAg, 29