

## SimpleGraphNoSql

Generated by Doxygen 1.8.14



# Contents

<b>1</b>	<b>Hierarchical Index</b>	<b>1</b>
1.1	Class Hierarchy . . . . .	1
<b>2</b>	<b>Class Index</b>	<b>3</b>
2.1	Class List . . . . .	3
<b>3</b>	<b>Class Documentation</b>	<b>5</b>
3.1	Tabela.Cell Class Reference . . . . .	5
3.1.1	Detailed Description . . . . .	6
3.1.2	Member Data Documentation . . . . .	6
3.1.2.1	data_boundary . . . . .	6
3.2	menus.Container Class Reference . . . . .	6
3.2.1	Detailed Description . . . . .	7
3.3	db.DB Class Reference . . . . .	7
3.3.1	Detailed Description . . . . .	7
3.3.2	Constructor & Destructor Documentation . . . . .	7
3.3.2.1	__init__() . . . . .	7
3.4	Trie.Nodo Class Reference . . . . .	8
3.4.1	Detailed Description . . . . .	8
3.4.2	Constructor & Destructor Documentation . . . . .	8
3.4.2.1	__init__() . . . . .	8
3.5	Tabela.RawTable Class Reference . . . . .	9
3.5.1	Detailed Description . . . . .	9
3.5.2	Constructor & Destructor Documentation . . . . .	9
3.5.2.1	__init__() . . . . .	9
3.6	Tabela.Table Class Reference . . . . .	10
3.6.1	Detailed Description . . . . .	10
3.6.2	Constructor & Destructor Documentation . . . . .	10
3.6.2.1	__init__() . . . . .	10
3.7	Trie.Trie Class Reference . . . . .	11
3.7.1	Detailed Description . . . . .	11
3.7.2	Constructor & Destructor Documentation . . . . .	11
3.7.2.1	__init__() . . . . .	11
	<b>Index</b>	<b>13</b>



# Chapter 1

## Hierarchical Index

### 1.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

object	
db.DB . . . . .	7
menus.Container . . . . .	6
Tabela.Cell . . . . .	5
Tabela.RawTable . . . . .	9
Tabela.Table . . . . .	10
Trie.Nodo . . . . .	8
Trie.Trie . . . . .	11



## Chapter 2

# Class Index

### 2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

<a href="#">Tabela.Cell</a>	5
<a href="#">menus.Container</a>	6
<a href="#">db.DB</a>	7
<a href="#">Trie.Nodo</a>	8
<a href="#">Tabela.RawTable</a>	9
<a href="#">Tabela.Table</a>	10
<a href="#">Trie.Trie</a>	11



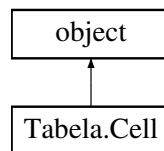


## Chapter 3

# Class Documentation

### 3.1 Tabela.Cell Class Reference

Inheritance diagram for Tabela.Cell:



#### Public Member Functions

- `def __init__(self, sheet, X, Y)`

#### Public Attributes

- [data\\_boundary](#)

*Inicializa celula, definindo atributos e o tipo da celula self.cell\_type define o tipo da variavel: Tipos possiveis: Leaf  
-> Representa uma celula do excel contendo dados e mais nada.*

- **blank\_boundary**
- **originx**
- **originy**
- **size\_x**
- **size\_y**
- **x**
- **y**
- **bounds**
- **cell\_type**
- **data**
- **table\_label**
- **parent\_node**
- **child\_nodes**
- **key\_row**
- **key\_col**
- **cell\_name**

### 3.1.1 Detailed Description

Classe que representa uma celula da tabela. Recebe um objeto do tipo sheet do xrlid como entrada e uma posicao X referente a linha e Y referente a celula. Cria um objeto do tipo celula classificando o ti

### 3.1.2 Member Data Documentation

#### 3.1.2.1 data\_boundary

Tabela.Cell.data\_boundary

Inicializa celula, definindo atributos e o tipo da celula self.cell\_type define o tipo da variavel: Tipos possiveis: Leaf  
-> Representa uma celula do excel contendo dados e mais nada.

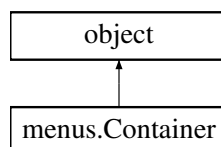
É relacionada a duas chaves, uma de linha(key) e outra de coluna(key\_col) : Key\_Row -> Representa uma chave que ordena outras chaves(exemplo: ANO). Uma por tabela Key\_Col ->Representa uma chave que dá sentido a uma coluna. Podem existir varias por tabela Key -> Representa uma chave presente em uma Key\_col Super\_Key -> Representa uma chave que é pai de outras chaves ou de uma chave coluna. Label -> Representa o nome da tabela, é a raiz da arvore e pai de todos os outros nodos Merge -> Representa uma celula que foi unida uma outra celula e não tem dados Blank ->Não faz parte da tabela, dado vazio que pode ser removido sem alterações

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

- C:/Users/PC/source/repos/Trabalho\_final\_cpd/SimpleGraphNoSQL/Tabela.py

## 3.2 menus.Container Class Reference

Inheritance diagram for menus.Container:



### Public Member Functions

- def `__init__`(self, data)

### Public Attributes

- raw\_type
- name
- data
- type

### 3.2.1 Detailed Description

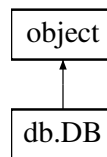
Classe representando um objeto que pode ser um `Nodo` ou uma `Tabela` ou uma `Lista` de `celulas` ou uma `Celula`. `data`

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

- `C:/Users/PC/source/repos/Trabalho_final_cpd/SimpleGraphNoSQL/menus.py`

## 3.3 db.DB Class Reference

Inheritance diagram for `db.DB`:



### Public Member Functions

- `def \_\_init\_\_(self, trie)`

### Public Attributes

- **`tables`**
- **`key_rows`**
- **`key_cols`**
- **`super_key`**

### 3.3.1 Detailed Description

Cria banco de dados a partir de uma `trie` de tabelas

### 3.3.2 Constructor & Destructor Documentation

#### 3.3.2.1 `__init__()`

```
def db.DB.__init__(  
    self,  
    trie )
```

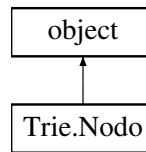
Inicialize banco de dados com uma `trie` recebida

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

- `C:/Users/PC/source/repos/Trabalho_final_cpd/SimpleGraphNoSQL/db.py`

## 3.4 Trie.Nodo Class Reference

Inheritance diagram for Trie.Nodo:



### Public Member Functions

- `def \_\_init\_\_(self, char_data, data)`

### Public Attributes

- **child**
- **parent**
- **chard**
- **data**

### 3.4.1 Detailed Description

Representa um nodo de uma trie

### 3.4.2 Constructor & Destructor Documentation

#### 3.4.2.1 `__init__()`

```
def Trie.Nodo.__init__ (
    self,
    char_data,
    data )
```

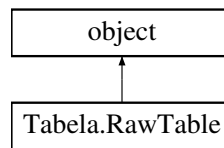
Inicializa nodos com dados. char\_data é o caractere, e data são os dados ligados aquele nodo. Representar nenh

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

- `C:/Users/PC/source/repos/Trabalho_final_cpd/SimpleGraphNoSQL/Trie.py`

## 3.5 Tabela.RawTable Class Reference

Inheritance diagram for Tabela.RawTable:



### Public Member Functions

- `def \_\_init\_\_(self, loc)`

### Public Attributes

- `loc_source`
- `raw_table`
- `raw_sheet`
- `table_label`
- `table_data`
- `bound_x`
- `bound_y`
- `key_cols`
- `key_row`
- `keys`

### 3.5.1 Detailed Description

Classe que representa uma tabela do excel na memoria. Guarda mais dados do que o necessario e não consegue ser

### 3.5.2 Constructor & Destructor Documentation

#### 3.5.2.1 `__init__()`

```
def Tabela.RawTable.__init__ (
    self,
    loc )
```

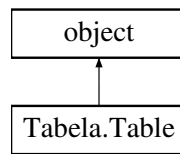
Inicializa uma tabela a partir de um local loc

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

- `C:/Users/PC/source/repos/Trabalho_final_cpd/SimpleGraphNoSQL/Tabela.py`

## 3.6 Tabela.Table Class Reference

Inheritance diagram for Tabela.Table:



### Public Member Functions

- `def \_\_init\_\_ (self, raw_table)`

### Public Attributes

- `loc_source`
- `table_label`
- `table_data`
- `bound_x`
- `bound_y`
- `key_cols`
- `key_row`
- `keys`

### 3.6.1 Detailed Description

Classe que representa uma tabela logica. Recebema uma Raw\_Table e remove os campos desnecessarios Pode ser pic

### 3.6.2 Constructor & Destructor Documentation

#### 3.6.2.1 `__init__()`

```
def Tabela.Table.__init__ (  
    self,  
    raw_table )
```

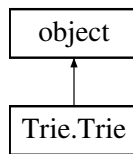
Inicializa Tabela com dados da raw table

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

- `C:/Users/PC/source/repos/Trabalho_final_cpd/SimpleGraphNoSQL/Tabela.py`

## 3.7 Trie.Trie Class Reference

Inheritance diagram for Trie.Trie:



### Public Member Functions

- `def \_\_init\_\_ (self)`
- `def yield\_strings (self, trie)`

### Public Attributes

- `root`
- `strings\_dict`
- `strings\_list`
- `reverse`

### 3.7.1 Detailed Description

Implementa uma Trie para pesquisa no nome de tabelas de maneira eficiente

### 3.7.2 Constructor & Destructor Documentation

#### 3.7.2.1 `__init__()`

```
def Trie.Trie.__init__ (  
    self )
```

Inicializa raiz da trie como um defaultdict com uma factory de dicionarios

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

- `C:/Users/PC/source/repos/Trabalho_final_cpd/SimpleGraphNoSQL/Trie.py`





# Index

- \_\_init\_\_
  - db::DB, [7](#)
  - Tabela::RawTable, [9](#)
  - Tabela::Table, [10](#)
  - Trie::Nodo, [8](#)
  - Trie::Trie, [11](#)
- data\_boundary
  - Tabela::Cell, [6](#)
- db.DB, [7](#)
- db::DB
  - \_\_init\_\_, [7](#)
- menus.Container, [6](#)
- Tabela.Cell, [5](#)
- Tabela.RawTable, [9](#)
- Tabela.Table, [10](#)
- Tabela::Cell
  - data\_boundary, [6](#)
- Tabela::RawTable
  - \_\_init\_\_, [9](#)
- Tabela::Table
  - \_\_init\_\_, [10](#)
- Trie.Nodo, [8](#)
- Trie.Trie, [11](#)
- Trie::Nodo
  - \_\_init\_\_, [8](#)
- Trie::Trie
  - \_\_init\_\_, [11](#)