

Sistema para Bibliotecas

AUTHOR

Versão 1.0

Quinta, 13 de Junho de 2019

Sumário

Table of contents

Índice Hierárquico

Hierarquia de Classes

Esta lista de hierarquias está parcialmente ordenada (ordem alfabética):

Busca	5
Cadastro.....	7
exception	
ExcecaoAberturaDeArquivo	9
ExcecaoAberturaDeArquivoLivros	10
ExcecaoAcesso.....	11
ExcecaoCadastraUsuario.....	12
ExcecaoComparaSenha	13
ExcecaoEntradaDeArquivo	14
ExcecaoFechamentoDeArquivo	15
ExcecaoFechamentoDeArquivoLivros	16
ExcecaoGetContraSenha	17
ExcecaoGetSenha.....	18
ExcecaoLeituraDeArquivoLivros	19
ExcecaoLivrosNaoListados.....	20
ExcecaoMesmoLivro	21
ExcecaoProblemaDevolucao.....	22
ExcecaoSetContato	23
ExcecaoSetCPF	24
ExcecaoSetEndereco	25
ExcecaoSetLevel	26
ExcecaoUsuarioNaoCadastrado	27
 Login	28
Menu	30
Pessoa.....	32
Usuario	38
 Senha	36

Índice dos Componentes

Lista de Classes

Aqui estão as classes, estruturas, uniões e interfaces e suas respectivas descrições:

Busca (Classe Busca é responsável por buscar os exemplares de Livros Cadastrados em um arquivo de texto e retornar o resultado na tela)	5
Cadastro (Classe Cadastro, tem a função de imprimir comandos em tela para o usuário e coletar as informações básicas para cadastramento do usuário)	7
ExcecaoAberturaDeArquivo (Classe ExcecaoAberturaDeArquivo retorna um erro caso ocorra um problema na abertura de um arquivo)	9
ExcecaoAberturaDeArquivoLivros (ExcecaoAberturaDeArquivoLivros retorna erro caso ocorra algum problema na listagem de livros)	10
ExcecaoAcesso (A classe ExcecaoAcesso retorna um erro caso a senha inserida pelo usuário não seja igual a senha a cadastrada)	11
ExcecaoCadastraUsuario	12
ExcecaoComparaSenha	13
ExcecaoEntradaDeArquivo (Classe ExcecaoEntradaDeArquivo retorna erro caso ocorra um problema na passagem das informações para um arquivo)	14
ExcecaoFechamentoDeArquivo (Classe ExcecaoFechamentoDeArquivo retorna um erro caso ocorra um problema no fechamento de um arquivo)	15
ExcecaoFechamentoDeArquivoLivros (ExcecaoFechamentoDeArquivoLivros retorna erro caso ocorra algum problema no fechamento da lista de livros)	16
ExcecaoGetContraSenha	17
ExcecaoGetSenha	18
ExcecaoLeituraDeArquivoLivros (ExcecaoLeituraDeArquivoLivros retorna erro caso ocorra erro na leitura do arquivo de listagem de livros)	19
ExcecaoLivrosNaoListados (ExcecaoLivrosNaoListados retorna erro caso o usuário busque por um livro que não esteja cadastrado no acervo)	20
ExcecaoMesmoLivro (Classe ExcecaoMesmoLivro retorna um erro caso o usuário tente realizar um empréstimo duplicado)	21
ExcecaoProblemaDevolucao (Classe ExcecaoProblemaDevolucao retorna um erro caso ocorra algum problema com a devolução do livro)	22
ExcecaoSetContato	23
ExcecaoSetCPF	24
ExcecaoSetEndereco	25
ExcecaoSetLevel	26
ExcecaoUsuarioNaoCadastrado (A classe ExcecaoUsuarioNaoCadastrado retorna um erro caso o usuário inserido não esteja presente no arquivo de autenticação)	27
Login (Classe Login, compara a linha referente a senha no arquivo de autenticação e a senha digitada pelo usuário)	28
Menu	30
Pessoa (A classe Pessoa é uma classe abstrata)	32
Senha (Classe Senha é responsável pela dupla comparação das senhas no momento do cadastramento)	36
Usuario	38

Índice dos Arquivos

Lista de Arquivos

Esta é a lista de todos os arquivos e suas respectivas descrições:

include/busca/Busca.hpp	40
include/cadastro/Cadastro.hpp	41
include/cadastro/Senha.hpp	42
include/excecoes/ExcecaoArquivos.hpp	43
include/excecoes/ExcecaoBusca.hpp	44
include/excecoes/ExcecaoCadastro.hpp	45
include/excecoes/ExcecaoLogin.hpp	46
include/excecoes/ExcecaoManipulacaoLivros.hpp	47
include/gestao/Pessoa.hpp	48
include/gestao/Usuario.hpp	49
include/login/Login.hpp	50
include/menu/Menu.hpp	51
src/busca/Busca.cpp	52
src/cadastro/Cadastro.cpp	53
src/cadastro/Senha.cpp	54
src/excecoes/ExcecaoArquivos.cpp	55
src/excecoes/ExcecaoBusca.cpp	56
src/excecoes/ExcecaoCadastro.cpp	57
src/excecoes/ExcecaoLogin.cpp	58
src/excecoes/ExcecaoManipulacaoLivros.cpp	59
src/gestao/Pessoa.cpp	60
src/gestao/Usuario.cpp	61
src/login/Login.cpp	62
src/menu/Menu.cpp	63

Classes

Referência da Classe Busca

Classe **Busca** é responsável por buscar os exemplares de Livros Cadastrados em um arquivo de texto e retornar o resultado na tela.

```
#include <Busca.hpp>
```

Membros Públicos

- `bool incluirLivros ()`
- `void buscarLivro (std::string palavra_chave)`
- `void printBusca ()`
- `std::vector< std::string > getLivrosBusca ()`

Descrição detalhada

Classe **Busca** é responsável por buscar os exemplares de Livros Cadastrados em um arquivo de texto e retornar o resultado na tela.

Funções membros

`void Busca::buscarLivro (std::string palavra_chave)`

Parâmetros:

<i>recebe</i>	uma string contendo o nome do livro desejado e procura em uma lista do acervo.
---------------	--

Exceções:

<i>std::runtime_error</i>	"Nao ha livros catalogados na biblioteca ou houve um erro na inclusao dos livros a partir do arquivo\nPor favor contactar o suporte"
---------------------------	--

`std::vector< std::string > Busca::getLivrosBusca ()`

`bool Busca::incluirLivros ()`

Parâmetros:

<i>Faz</i>	a leitura de um arquivo de texto com o acervo de livros e o armazena em um vetor de strings.
------------	--

Retorna:

true caso o armazenamento dos livros no vetor tenha ocorrido com sucesso.

Exceções:

<i>std::runtime_error</i>	"Houve um problema com a abertura da listagem de livros\nPor favor contactar o suporte"
<i>std::runtime_error</i>	"Houve um problema com o fechamento da listagem de livros\nPor favor contactar o suporte"
<i>std::runtime_error</i>	"Houve um problema com a leitura da listagem de livros\nPor favor contactar o suporte"

void Busca::printBusca ()

Parâmetros:

<i>Método</i>	buscarLivro.
---------------	--------------

Retorna:

Retorna na tela do usuário os resultados da busca realizada.

A documentação para essa classe foi gerada a partir dos seguintes arquivos:

- include/busca/**Busca.hpp**
- src/busca/**Busca.cpp**

Referência da Classe Cadastro

Classe **Cadastro**, tem a função de imprimir comandos em tela para o usuário e coletar as informações básicas para cadastramento do usuário.

```
#include <Cadastro.hpp>
```

Membros Públicos

- `bool setNome ()`
- `bool setCpf ()`
- `bool setEndereco ()`
- `bool setContato ()`
- `bool cadastrarPessoa ()`

Descrição detalhada

Classe **Cadastro**, tem a função de imprimir comandos em tela para o usuário e coletar as informações básicas para cadastramento do usuário.

Funções membros

`bool Cadastro::cadastrarPessoa ()`

Parâmetros:

<i>nome.</i>	
<i>cpf.</i>	
<i>endereco</i>	
<i>contato</i>	

Retorna:

true caso o cadastro tenha sido bem sucedido, caso contrário, retornará false.

Exceções:

<i>std::runtime_error</i>	"Houve um problema com o cadastramento do usuario".
---------------------------	---

`bool Cadastro::setContato ()`

Parâmetros:

<i>recebe</i>	o contato inserido pelo usuário.
---------------	----------------------------------

Exceções:

<i>std::runtime_error</i>	"Houve um problema com o contato do usuario".
---------------------------	---

`bool Cadastro::setCpf ()`

Parâmetros:

<i>recebe</i>	o cpf inserido pelo usuário.
---------------	------------------------------

Exceções:

<i>std::runtime_error</i>	"Houve um problema com o CPF inserido".
---------------------------	---

bool Cadastro::setEndereco ()

Parâmetros:

<i>recebe</i>	o endereço inserido pelo usuário.
---------------	-----------------------------------

Exceções:

<i>std::runtime_error</i>	"Houve um problema com o endereco do usuario".
---------------------------	--

bool Cadastro::setNome ()

Parâmetros:

<i>recebe</i>	o nome inserido pelo usuário.
---------------	-------------------------------

Exceções:

<i>std::runtime_error</i>	"Houve um problema com o nome inserido".
---------------------------	--

A documentação para essa classe foi gerada a partir dos seguintes arquivos:

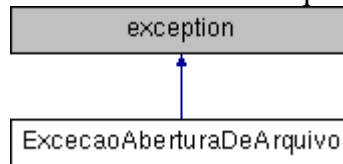
- include/cadastro/**Cadastro.hpp**
- src/cadastro/**Cadastro.cpp**

Referência da Classe ExcecaoAberturaDeArquivo

Classe **ExcecaoAberturaDeArquivo** retorna um erro caso ocorra um problema na abertura de um arquivo.

```
#include <ExcecaoArquivos.hpp>
```

Diagrama de hierarquia para ExcecaoAberturaDeArquivo:



Membros Públicos

- virtual const char * **what** () const throw ()

Descrição detalhada

Classe **ExcecaoAberturaDeArquivo** retorna um erro caso ocorra um problema na abertura de um arquivo.

Funções membros

const char * ExcecaoAberturaDeArquivo::what () const throw () [virtual]

Exceções:

<i>std::runtime_error</i>	"Houve um problema com a abertura do seu registro. Por favor contactar o suporte"
---------------------------	---

A documentação para essa classe foi gerada a partir dos seguintes arquivos:

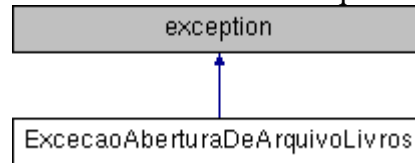
- include/excecoes/**ExcecaoArquivos.hpp**
- src/excecoes/**ExcecaoArquivos.cpp**

Referência da Classe ExcecaoAberturaDeArquivoLivros

ExcecaoAberturaDeArquivoLivros retorna erro caso ocorra algum problema na listagem de livros.

#include <ExcecaoBusca.hpp>

Diagrama de hierarquia para ExcecaoAberturaDeArquivoLivros:



Membros Públicos

- virtual const char * **what** () const throw ()

Descrição detalhada

ExcecaoAberturaDeArquivoLivros retorna erro caso ocorra algum problema na listagem de livros.

Funções membros

const char * ExcecaoAberturaDeArquivoLivros::what () const throw () [virtual]

Exceções:

<i>std::runtime_error</i>	"Houve um problema com a abertura da listagem de livros\nPor favor contactar o suporte"
---------------------------	---

A documentação para essa classe foi gerada a partir dos seguintes arquivos:

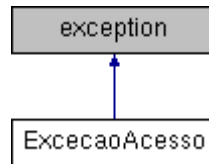
- include/excecoes/**ExcecaoBusca.hpp**
- src/excecoes/**ExcecaoBusca.cpp**

Referência da Classe ExcecaoAcesso

A classe **ExcecaoAcesso** retorna um erro caso a senha inserida pelo usuário não seja igual a senha a cadastrada.

```
#include <ExcecaoLogin.hpp>
```

Diagrama de hierarquia para ExcecaoAcesso:



Membros Públicos

- `virtual const char * what () const throw ()`

Descrição detalhada

A classe **ExcecaoAcesso** retorna um erro caso a senha inserida pelo usuário não seja igual a senha a cadastrada.

Funções membros

`const char * ExcecaoAcesso::what () const throw () [virtual]`

Exceções:

<code>std::runtime_error</code>	"ACESSO NEGADO. Por favor confira se digitou a senha corretamente"
---------------------------------	--

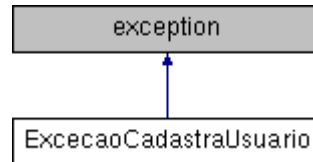
A documentação para essa classe foi gerada a partir dos seguintes arquivos:

- `include/excecoes/ExcecaoLogin.hpp`
- `src/excecoes/ExcecaoLogin.cpp`

Referência da Classe ExcecaoCadastraUsuario

```
#include <ExcecaoCadastro.hpp>
```

Diagrama de hierarquia para ExcecaoCadastraUsuario:



Membros Públicos

- virtual const char * **what** () const throw ()

Funções membros

const char * ExcecaoCadastraUsuario::what () const throw () [virtual]

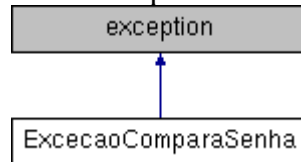
A documentação para essa classe foi gerada a partir dos seguintes arquivos:

- include/excecoes/**ExcecaoCadastro.hpp**
- src/excecoes/**ExcecaoCadastro.cpp**

Referência da Classe ExcecaoComparaSenha

```
#include <ExcecaoCadastro.hpp>
```

Diagrama de hierarquia para ExcecaoComparaSenha:



Membros Públicos

- `virtual const char * what () const throw ()`

Funções membros

`const char * ExcecaoComparaSenha::what () const throw () [virtual]`

A documentação para essa classe foi gerada a partir dos seguintes arquivos:

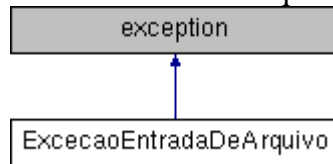
- `include/excecoes/ExcecaoCadastro.hpp`
- `src/excecoes/ExcecaoCadastro.cpp`

Referência da Classe ExcecaoEntradaDeArquivo

Classe **ExcecaoEntradaDeArquivo** retorna erro caso ocorra um problema na passagem das informações para um arquivo.

```
#include <ExcecaoArquivos.hpp>
```

Diagrama de hierarquia para ExcecaoEntradaDeArquivo:



Membros Públicos

- virtual const char * **what** () const throw ()

Descrição detalhada

Classe **ExcecaoEntradaDeArquivo** retorna erro caso ocorra um problema na passagem das informações para um arquivo.

Funções membros

const char * ExcecaoEntradaDeArquivo::what () const throw () [virtual]

Exceções:

<i>std::runtime_error</i>	"Houve um problema com o conteúdo do seu registro.Por favor contactar o suporte".
---------------------------	---

A documentação para essa classe foi gerada a partir dos seguintes arquivos:

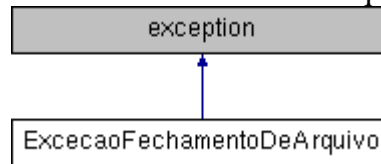
- include/excecoes/**ExcecaoArquivos.hpp**
- src/excecoes/**ExcecaoArquivos.cpp**

Referência da Classe ExcecaoFechamentoDeArquivo

Classe **ExcecaoFechamentoDeArquivo** retorna um erro caso ocorra um problema no fechamento de um arquivo.

```
#include <ExcecaoArquivos.hpp>
```

Diagrama de hierarquia para **ExcecaoFechamentoDeArquivo**:



Membros Públicos

- virtual const char * **what** () const throw ()

Descrição detalhada

Classe **ExcecaoFechamentoDeArquivo** retorna um erro caso ocorra um problema no fechamento de um arquivo.

Funções membros

const char * ExcecaoFechamentoDeArquivo::what () const throw () [virtual]

Exceções:

<i>std::runtime_error</i>	"Houve um problema com o acesso do seu registro\nPor favor contactar o suporte".
---------------------------	--

A documentação para essa classe foi gerada a partir dos seguintes arquivos:

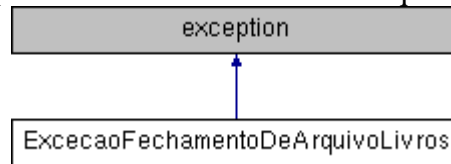
- include/excecoes/**ExcecaoArquivos.hpp**
- src/excecoes/**ExcecaoArquivos.cpp**

Referência da Classe ExcecaoFechamentoDeArquivoLivros

ExcecaoFechamentoDeArquivoLivros retorna erro caso ocorra algum problema no fechamento da lista de livros.

```
#include <ExcecaoBusca.hpp>
```

Diagrama de hierarquia para ExcecaoFechamentoDeArquivoLivros:



Membros Públicos

- virtual const char * **what** () const throw ()

Descrição detalhada

ExcecaoFechamentoDeArquivoLivros retorna erro caso ocorra algum problema no fechamento da lista de livros.

Funções membros

```
const char * ExcecaoFechamentoDeArquivoLivros::what () const throw () [virtual]
```

Exceções:

<i>std::runtime_error</i>	"Houve um problema com o fechamento da listagem de livros\nPor favor contactar o suporte"
---------------------------	---

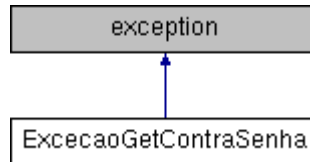
A documentação para essa classe foi gerada a partir dos seguintes arquivos:

- include/excecoes/**ExcecaoBusca.hpp**
- src/excecoes/**ExcecaoBusca.cpp**

Referência da Classe ExcecaoGetContraSenha

```
#include <ExcecaoCadastro.hpp>
```

Diagrama de hierarquia para ExcecaoGetContraSenha:



Membros Públicos

- `virtual const char * what () const throw ()`

Funções membros

`virtual const char* ExcecaoGetContraSenha::what () const throw () [virtual]`

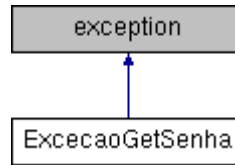
A documentação para essa classe foi gerada a partir do seguinte arquivo:

- `include/excecoes/ExcecaoCadastro.hpp`

Referência da Classe ExcecaoGetSenha

```
#include <ExcecaoCadastro.hpp>
```

Diagrama de hierarquia para ExcecaoGetSenha:



Membros Públicos

- virtual const char * **what** () const throw ()

Funções membros

virtual const char* ExcecaoGetSenha::what () const throw () [virtual]

A documentação para essa classe foi gerada a partir do seguinte arquivo:

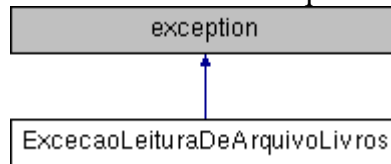
- include/excecoes/**ExcecaoCadastro.hpp**

Referência da Classe ExcecaoLeituraDeArquivoLivros

ExcecaoLeituraDeArquivoLivros retorna erro caso ocorra erro na leitura do arquivo de listagem de livros.

```
#include <ExcecaoBusca.hpp>
```

Diagrama de hierarquia para **ExcecaoLeituraDeArquivoLivros**:



Membros Públicos

- virtual const char * **what** () const throw ()

Descrição detalhada

ExcecaoLeituraDeArquivoLivros retorna erro caso ocorra erro na leitura do arquivo de listagem de livros.

Funções membros

const char * ExcecaoLeituraDeArquivoLivros::what () const throw () [virtual]

Exceções:

<i>std::runtime_error</i>	"Houve um problema com a leitura da listagem de livros\nPor favor contactar o suporte"
---------------------------	--

A documentação para essa classe foi gerada a partir dos seguintes arquivos:

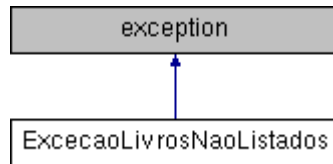
- include/excecoes/**ExcecaoBusca.hpp**
- src/excecoes/**ExcecaoBusca.cpp**

Referência da Classe ExcecaoLivrosNaoListados

ExcecaoLivrosNaoListados retorna erro caso o usuário busque por um livro que não esteja cadastrado no acervo.

#include <ExcecaoBusca.hpp>

Diagrama de hierarquia para ExcecaoLivrosNaoListados:



Membros Públicos

- virtual const char * **what** () const throw ()

Descrição detalhada

ExcecaoLivrosNaoListados retorna erro caso o usuário busque por um livro que não esteja cadastrado no acervo.

Funções membros

const char * ExcecaoLivrosNaoListados::what () const throw () [virtual]

Exceções:

<i>std::runtime_error</i>	"Nao ha livros catalogados na biblioteca ou houve um erro na inclusao dos livros a partir do arquivo. Por favor contactar o suporte".
---------------------------	---

A documentação para essa classe foi gerada a partir dos seguintes arquivos:

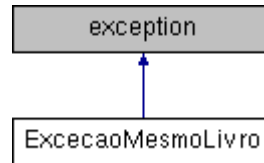
- include/excecoes/**ExcecaoBusca.hpp**
- src/excecoes/**ExcecaoBusca.cpp**

Referência da Classe ExcecaoMesmoLivro

Classe **ExcecaoMesmoLivro** retorna um erro caso o usuário tente realizar um empréstimo duplicado.

#include <ExcecaoManipulacaoLivros.hpp>

Diagrama de hierarquia para ExcecaoMesmoLivro:



Membros Públicos

- virtual const char * **what** () const throw ()

Descrição detalhada

Classe **ExcecaoMesmoLivro** retorna um erro caso o usuário tente realizar um empréstimo duplicado.

Funções membros

const char * ExcecaoMesmoLivro::what () const throw () [virtual]

Exceções:

<code>std::runtime_error</code>	"O usuario ja esta com esse livro emprestado"
---------------------------------	---

A documentação para essa classe foi gerada a partir dos seguintes arquivos:

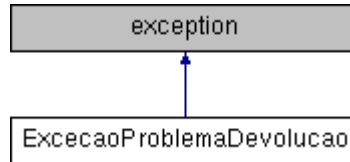
- include/excecoes/**ExcecaoManipulacaoLivros.hpp**
- src/excecoes/**ExcecaoManipulacaoLivros.cpp**

Referência da Classe ExcecaoProblemaDevolucao

Classe **ExcecaoProblemaDevolucao** retorna um erro caso ocorra algum problema com a devolução do livro.

```
#include <ExcecaoManipulacaoLivros.hpp>
```

Diagrama de hierarquia para **ExcecaoProblemaDevolucao**:



Membros Públicos

- virtual const char * **what** () const throw ()

Descrição detalhada

Classe **ExcecaoProblemaDevolucao** retorna um erro caso ocorra algum problema com a devolução do livro.

Funções membros

const char * **ExcecaoProblemaDevolucao::what** () const throw () [virtual]

Exceções:

<code>std::runtime_error</code>	"Houve um problema com a devolucao do Livro".
---------------------------------	---

A documentação para essa classe foi gerada a partir dos seguintes arquivos:

- include/excecoes/**ExcecaoManipulacaoLivros.hpp**
- src/excecoes/**ExcecaoManipulacaoLivros.cpp**

Referência da Classe ExcecaoSetContato

```
#include <ExcecaoCadastro.hpp>
```

Diagrama de hierarquia para ExcecaoSetContato:



Membros Públicos

- `virtual const char * what () const throw ()`

Funções membros

`const char * ExcecaoSetContato::what () const throw () [virtual]`

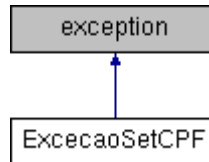
A documentação para essa classe foi gerada a partir dos seguintes arquivos:

- `include/excecoes/ExcecaoCadastro.hpp`
- `src/excecoes/ExcecaoCadastro.cpp`

Referência da Classe ExcecaoSetCPF

```
#include <ExcecaoCadastro.hpp>
```

Diagrama de hierarquia para ExcecaoSetCPF:



Membros Públicos

- `virtual const char * what () const throw ()`

Funções membros

`const char * ExcecaoSetCPF::what () const throw () [virtual]`

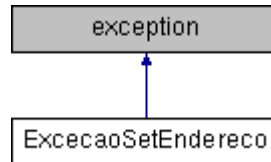
A documentação para essa classe foi gerada a partir dos seguintes arquivos:

- `include/excecoes/ExcecaoCadastro.hpp`
- `src/excecoes/ExcecaoCadastro.cpp`

Referência da Classe ExcecaoSetEndereco

```
#include <ExcecaoCadastro.hpp>
```

Diagrama de hierarquia para ExcecaoSetEndereco:



Membros Públicos

- `virtual const char * what () const throw ()`

Funções membros

`const char * ExcecaoSetEndereco::what () const throw () [virtual]`

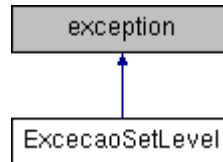
A documentação para essa classe foi gerada a partir dos seguintes arquivos:

- `include/excecoes/ExcecaoCadastro.hpp`
- `src/excecoes/ExcecaoCadastro.cpp`

Referência da Classe ExcecaoSetLevel

```
#include <ExcecaoCadastro.hpp>
```

Diagrama de hierarquia para ExcecaoSetLevel:



Membros Públicos

- `virtual const char * what () const throw ()`

Funções membros

`const char * ExcecaoSetLevel::what () const throw () [virtual]`

A documentação para essa classe foi gerada a partir dos seguintes arquivos:

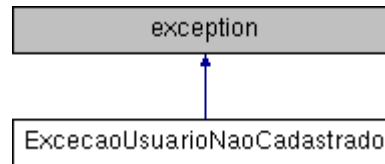
- `include/excecoes/ExcecaoCadastro.hpp`
- `src/excecoes/ExcecaoCadastro.cpp`

Referência da Classe ExcecaoUsuarioNaoCadastrado

A classe **ExcecaoUsuarioNaoCadastrado** retorna um erro caso o usuario inserido não esteja presente no arquivo de autenticação.

```
#include <ExcecaoLogin.hpp>
```

Diagrama de hierarquia para ExcecaoUsuarioNaoCadastrado:



Membros Públicos

- virtual const char * **what** () const throw ()

Descrição detalhada

A classe **ExcecaoUsuarioNaoCadastrado** retorna um erro caso o usuario inserido não esteja presente no arquivo de autenticação.

Funções membros

const char * ExcecaoUsuarioNaoCadastrado::what () const throw () [virtual]

Exceções:

<i>std::runtime_error</i>	"ACESSO NEGADO. O usuario apresentado nao esta cadastrado. Por favor tente novamente ou cadastre um novo usuario"
---------------------------	---

A documentação para essa classe foi gerada a partir dos seguintes arquivos:

- include/excecoes/**ExcecaoLogin.hpp**
- src/excecoes/**ExcecaoLogin.cpp**

Referência da Classe Login

Classe **Login**, compara a linha referente a senha no arquivo de autenticação e a senha digitada pelo usuário.

```
#include <Login.hpp>
```

Membros Públicos

- `std::string getCpf ()`
- `bool autenticacao ()`

Atributos Públicos

- `std::string usuarioTentativa`
- `std::string senhaTentativa`

Descrição detalhada

Classe **Login**, compara a linha referente a senha no arquivo de autenticação e a senha digitada pelo usuário.

Funções membros

bool Login::autenticacao ()

Compara a senha cadastrada, presente no arquivo autenticação, com a senha inserida pelo usuário.

Parâmetros:

<i>senhaArquivo, armazena</i>	a informação contida no arquivo
<i>senhaTentativa, é</i>	a senha inserida pelo usuário.

Retorna:

true se as senhas forem idênticas, false, se ao contrário.

Exceções:

<i>std::runtime_error</i>	"ACESSO NEGADO Por favor confira se digitou a senha corretamente"
---------------------------	---

std::string Login::getCpf ()

Parâmetros:

<i>Pega</i>	o numero do cpf no arquivo de autenticação que serve como autenticador de usuário.
-------------	--

Retorna:

valor do cpf

Exceções:

<i>std::runtime_error</i>	"ACESSO NEGADO\nO usuario apresentado nao esta cadastrado\nPor favor tente novamente ou cadastre um novo usuario"
---------------------------	---

Atributos

std::string Login::senhaTentativa

std::string Login::usuarioTentativa

A documentação para essa classe foi gerada a partir dos seguintes arquivos:

- `include/login/Login.hpp`
- `src/login/Login.cpp`

Referência da Classe Menu

```
#include <Menu.hpp>
```

Membros Públicos

- bool **cabecalho** ()
- bool **textoLogin** ()
- bool **textoMenu** ()
- bool **textoBusca** ()
- bool **textoTeclaErrada** ()
- bool **systemPause** ()

Descrição detalhada

A classe **Menu** possui métodos exclusivamente dedicados à impressão de informações na tela do usuário.

Funções membros

bool Menu::cabecalho ()

Imprime o cabeçalho do sistema.

Retorna:

texto contendo o cabeçalho do sistema.

bool Menu::systemPause ()

Tem função de parar o programa até que alguma tecla seja pressionada.

Parâmetros:

<i>Qualquer</i>	tecla pode ser pressionada
-----------------	----------------------------

Retorna:

Fim do método e continuação no fluxo do sistema.

bool Menu::textoBusca ()

Imprime texto referente a busca.

Retorna:

informações necessárias para realização das buscas.

bool Menu::textoLogin ()

Imprime textos referente as opção de login.

Retorna:

texto de comando para login, cadastramente e saída.

bool Menu::textoMenu ()

Imprime texto referente as opção de emprestimo.

Retorna:

texto de comando para busca, emprestimo/devolução e saída.

bool Menu::textoTeclaErrada ()

Imprime aviso de digitação inválida.

Retorna:

Informação para conscientização do usuário.

A documentação para essa classe foi gerada a partir dos seguintes arquivos:

- include/menu/**Menu.hpp**
- src/menu/**Menu.cpp**

Referência da Classe Pessoa

A classe **Pessoa** é uma classe abstrata.

```
#include <Pessoa.hpp>
```

Diagrama de hierarquia para Pessoa:



Membros Públicos

- **Pessoa** (std::string cpf)
- virtual ~**Pessoa** ()
- bool **buscarDados** ()
- std::string **getNome** ()
- std::string **getCpf** ()
- std::string **getEndereco** ()
- std::string **getContato** ()
- virtual bool **verificarSeHaLivrosEmprestado** ()=0
- virtual bool **pegarLivroEmprestado** (std::string)=0
- virtual bool **devolverLivro** (std::string)=0
- virtual bool **printLivrosEmprestados** ()=0
- virtual std::vector< std::string > **getLivros** ()=0
- virtual bool **verificarLivroEspecifico** (std::string)=0

Descrição detalhada

A classe **Pessoa** é uma classe abstrata.

Construtores e Destrutores

Pessoa::Pessoa (std::string *cpf*)

Construtor simples.

Parâmetros:

<i>recebe</i>	o cpf e setar o atributo da classe.
---------------	-------------------------------------

Retorna:

@Pessoa

Exceções:

--	--

Pessoa::~~Pessoa ()[virtual]

O destrutor deve ser virtual uma vez que a classe é abstrata e haverá substituição de Liskov em **Usuario**.

Funções membros

bool Pessoa::buscarDados ()

Seta todos os atributos com as informações contidas em arquivo. O tipo booleano é para que fosse possível atender aos testes.

Parâmetros:

<i>nome</i>	
<i>cpf</i>	
<i>endereço</i>	
<i>contato</i>	

Exceções:

<i>std::runtime_error</i>	"Houve um problema com a abertura do seu registro\nPor favor contactar o suporte"
<i>std::runtime_error</i>	"Houve um problema com o conteúdo do seu registro\nPor favor contactar o suporte"

virtual bool Pessoa::devolverLivro (std::string) [pure virtual]

Método Virtual Puro

Parâmetros:

<i>Possui</i>	uma string como parametro, no caso, o nome do livro
---------------	---

Retorna:

Retorna true caso consigo devolver um livro e falso caso contrário.
Implementado por **Usuario** (p.38).

std::string Pessoa::getContato ()

Seta o atributo contato através das informações contidas em arquivo.

Parâmetros:

<i>contato</i>	no arquivo
----------------	------------

Retorna:

retorna o contato no atributo privado `_contato`.

std::string Pessoa::getCpf ()

Seta o atributo cpf através das informações contidas em arquivo.

Parâmetros:

<i>cpf</i>	no arquivo
------------	------------

Retorna:

retorna o cpf no atributo privado `_cpf`.

std::string Pessoa::getEndereco ()

Seta o atributo endereço através das informações contidas em arquivo.

Parâmetros:

<i>endereço</i>	no arquivo
-----------------	------------

Retorna:

retorna o endereço no atributo privado `_endereco`.

virtual std::vector<std::string> Pessoa::getLivros () [pure virtual]

Retorna:

um vetor de string que ordena e enumera os livros emprestados para passagem de parametros em outro métodos descritos anteriormente.

Implementado por **Usuario** (p.38).

std::string Pessoa::getNome ()

Seta o atributo nome atraves das informações contidas em arquivo.

Parâmetros:

<i>nome</i>	no arquivo
-------------	------------

Retorna:

retorna o nome no atributo privado _nome.

virtual bool Pessoa::pegarLivroEmprestado (std::string) [pure virtual]

Método Virtual Puro

Parâmetros:

<i>Possui</i>	uma string como parametro, no caso, o nome do livro
---------------	---

Retorna:

Retorna true caso consigo colocar emprestar um livro e falso caso contrário.

Implementado por **Usuario** (p.38).

virtual bool Pessoa::printLivrosEmprestados () [pure virtual]

Método Virtual Puro

Retorna:

Retorna na tela todos os livros em posse do usuário, caso o usuario deseje devolver algum livro, este livro será passado como parametro no metodo devolverLivro.

Implementado por **Usuario** (p.39).

virtual bool Pessoa::verificarLivroEspecifico (std::string) [pure virtual]

Método Virtual Puro

Parâmetros:

<i>Possui</i>	uma string como parametro, no caso, o nome do livro
---------------	---

Retorna:

Retorna true caso o livro passado esteja emprestado para a pessoa que chamou o metodo e falso caso contrario.

Implementado por **Usuario** (p.39).

virtual bool Pessoa::verificarSeHaLivrosEmprestado () [pure virtual]

Método Virtual Puro

Parâmetros:

<i>cpf</i>	do usuario e verifica se o mesmo possui algum emprestimo
------------	--

Retorna:

true se existir emprestimo e false caso negativo

Implementado por **Usuario** (p.39).

A documentação para essa classe foi gerada a partir dos seguintes arquivos:

- include/gestao/**Pessoa.hpp**
- src/gestao/**Pessoa.cpp**

Referência da Classe Senha

Classe **Senha** é responsável pela dupla comparação das senhas no momento do cadastramento.

```
#include <Senha.hpp>
```

Membros Públicos

- void **setSenha** ()
 - void **setContraSenha** ()
 - bool **comparaSenha** ()
 - bool **cadastraSenha** ()
-

Descrição detalhada

Classe **Senha** é responsável pela dupla comparação das senhas no momento do cadastramento.

Funções membros

bool Senha::cadastraSenha ()

Parâmetros:

<i>resultado</i>	do método comparaSenha (). Em caso afirmativo, a senha é armazenada no arquivo autenticação.
------------------	---

bool Senha::comparaSenha ()

Parâmetros:

<i>senha</i>	inserida
<i>contra</i>	senha inserida

Retorna:

true caso a comparação seja idêntica e false em caso de divergências

Exceções:

<i>std::runtime_error</i>	"As senhas não conferem"
---------------------------	--------------------------

void Senha::setContraSenha ()

Parâmetros:

<i>recebe</i>	novamente a informação inserida pelo usuário, utilizada para confirmação através de comparação.
---------------	---

void Senha::setSenha ()

Parâmetros:

<i>recebe</i>	a informação inserida pelo usuário.
---------------	-------------------------------------

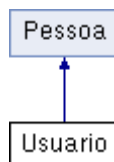
A documentação para essa classe foi gerada a partir dos seguintes arquivos:

- `include/cadastro/Senha.hpp`
- `src/cadastro/Senha.cpp`

Referência da Classe Usuario

```
#include <Usuario.hpp>
```

Diagrama de hierarquia para Usuario:



Membros Públicos

- **Usuario** (std::string)
- bool **verificarSeHaLivrosEmprestado** () override
- bool **verificarLivroEspecifico** (std::string) override
- bool **pegarLivroEmprestado** (std::string) override
- bool **devolverLivro** (std::string) override
- bool **printLivrosEmprestados** () override
- std::vector< std::string > **getLivros** () override

Construtores e Destrutores

Usuario::Usuario (std::string *cpf*)

Funções membros

bool Usuario::devolverLivro (std::string) [override], [virtual]

Método Virtual Puro

Parâmetros:

<i>Possui</i>	uma string como parametro, no caso, o nome do livro
---------------	---

Retorna:

Retorna true caso consigo devolver um livro e falso caso contrário.
Implementa **Pessoa** (p.33).

std::vector< std::string > Usuario::getLivros () [override], [virtual]

Retorna:

um vetor de string que ordena e enumera os livros emprestados para passagem de parametros em outro métodos descritos anteriormente.
Implementa **Pessoa** (p.33).

bool Usuario::pegarLivroEmprestado (std::string) [override], [virtual]

Método Virtual Puro

Parâmetros:

<i>Possui</i>	uma string como parametro, no caso, o nome do livro
---------------	---

Retorna:

Retorna true caso consigo colocar emprestar um livro e falso caso contrário.
Implementa **Pessoa** (p.34).

bool Usuario::printLivrosEmprestados () [override], [virtual]

Método Virtual Puro

Retorna:

Retorna na tela todos os livros em posse do usuário, caso o usuario deseje devolver algum livro, este livro será passado como parametro no metodo devolverLivro.

Implementa **Pessoa** (p.34).

bool Usuario::verificarLivroEspecifico (std::string) [override], [virtual]

Método Virtual Puro

Parâmetros:

<i>Possui</i>	uma string como parametro, no caso, o nome do livro
---------------	---

Retorna:

Retorna true caso o livro passado esteja emprestado para a pessoa que chamou o metodo e falso caso contrario.

Implementa **Pessoa** (p.34).

bool Usuario::verificarSeHaLivrosEmprestado () [override], [virtual]

Método Virtual Puro

Parâmetros:

<i>cpf</i>	do usuario e verifica se o mesmo possui algum emprestimo
------------	--

Retorna:

true se existir emprestimo e false caso negativo

Implementa **Pessoa** (p.34).

A documentação para essa classe foi gerada a partir dos seguintes arquivos:

- include/gestao/**Usuario.hpp**
- src/gestao/**Usuario.cpp**

Arquivos

Referência do Arquivo include/busca/Busca.hpp

```
#include <iostream>
#include <fstream>
#include <string>
#include <vector>
#include <algorithm>
#include <exception>
#include "../excecoes/ExcecaoBusca.hpp"
```

Componentes

- class **Busca**
*Classe **Busca** é responsável por buscar os exemplares de Livros Cadastrados em um arquivo de texto e retornar o resultado na tela.*

Referência do Arquivo include/cadastro/Cadastro.hpp

```
#include "../excecoes/ExcecaoArquivos.hpp"
#include <iostream>
#include <string>
#include <fstream>
```

Componentes

- class **Cadastro**
*Classe **Cadastro**, tem a função de imprimir comandos em tela para o usuário e coletar as informações básicas para cadastramento do usuário.*

Referência do Arquivo include/cadastro/Senha.hpp

```
#include "../excecoes/ExcecaoArquivos.hpp"  
#include <iostream>  
#include <string>  
#include <fstream>
```

Componentes

- class **Senha**
*Classe **Senha** é responsável pela dupla comparação das senhas no momento do cadastramento.*

Referência do Arquivo

include/excecoes/ExcecaoArquivos.hpp

```
#include <exception>
```

Componentes

- class **ExcecaoAberturaDeArquivo**
*Classe **ExcecaoAberturaDeArquivo** retorna um erro caso ocorra um problema na abertura de um arquivo.*
- class **ExcecaoFechamentoDeArquivo**
*Classe **ExcecaoFechamentoDeArquivo** retorna um erro caso ocorra um problema no fechamento de um arquivo.*
- class **ExcecaoEntradaDeArquivo**
*Classe **ExcecaoEntradaDeArquivo** retorna erro caso ocorra um problema na passagem das informações para um arquivo.*

Referência do Arquivo include/excecoes/ExcecaoBusca.hpp

```
#include <exception>
```

Componentes

- class **ExcecaoLivrosNaoListados**
ExcecaoLivrosNaoListados retorna erro caso o usuário busque por um livro que não esteja cadastrado no acervo.
- class **ExcecaoAberturaDeArquivoLivros**
ExcecaoAberturaDeArquivoLivros retorna erro caso ocorra algum problema na listagem de livros.
- class **ExcecaoFechamentoDeArquivoLivros**
ExcecaoFechamentoDeArquivoLivros retorna erro caso ocorra algum problema no fechamento da lista de livros.
- class **ExcecaoLeituraDeArquivoLivros**
ExcecaoLeituraDeArquivoLivros retorna erro caso ocorra erro na leitura do arquivo de listagem de livros.

Referência do Arquivo

include/excecoes/ExcecaoCadastro.hpp

```
#include <exception>
```

Componentes

- class **ExcecaoCadastraUsuario**
- class **ExcecaoGetSenha**
- class **ExcecaoGetContraSenha**
- class **ExcecaoComparaSenha**
- class **ExcecaoSetCPF**
- class **ExcecaoSetEndereco**
- class **ExcecaoSetContato**
- class **ExcecaoSetLevel**

Referência do Arquivo include/excecoes/ExcecaoLogin.hpp

```
#include <exception>
```

Componentes

- class **ExcecaoAcesso**

*A classe **ExcecaoAcesso** retorna um erro caso a senha inserida pelo usuário não seja igual a senha a cadastrada.*

- class **ExcecaoUsuarioNaoCadastrado**

*A classe **ExcecaoUsuarioNaoCadastrado** retorna um erro caso o usuário inserido não esteja presente no arquivo de autenticação.*

Referência do Arquivo

include/excecoes/ExcecaoManipulacaoLivros.hpp

```
#include <exception>
```

Componentes

- class **ExcecaoMesmoLivro**
*Classe **ExcecaoMesmoLivro** retorna um erro caso o usuário tente realizar um empréstimo duplicado.*
- class **ExcecaoProblemaDevolucao**
*Classe **ExcecaoProblemaDevolucao** retorna um erro caso ocorra algum problema com a devolução do livro.*

Referência do Arquivo include/gestao/Pessoa.hpp

```
#include "../excecoes/ExcecaoArquivos.hpp"
#include <iostream>
#include <string>
#include <fstream>
#include <vector>
```

Componentes

- class **Pessoa**
*A classe **Pessoa** é uma classe abstrata.*

Referência do Arquivo include/gestao/Usuario.hpp

```
#include "Pessoa.hpp"  
#include "../excecoes/ExcecaoManipulacaoLivros.hpp"
```

Componentes

- class **Usuario**

Referência do Arquivo include/login/Login.hpp

```
#include <iostream>
#include <string>
#include <fstream>
```

Componentes

- class **Login**
*Classe **Login**, compara a linha referente a senha no arquivo de autenticação e a senha digitada pelo usuário.*

Referência do Arquivo include/menu/Menu.hpp

```
#include <iostream>
```

Componentes

- class **Menu**

Referência do Arquivo src/busca/Busca.cpp

```
#include "../..//include/busca/Busca.hpp"
```

Referência do Arquivo src/cadastro/Cadastro.cpp

```
#include "../..//include/cadastro/Cadastro.hpp"
```


Referência do Arquivo src/cadastro/Senha.cpp

```
#include "../..//include/cadastro/Senha.hpp"  
#include "../..//include/excecoes/ExcecaoArquivos.hpp"
```

Referência do Arquivo src/excecoes/ExcecaoArquivos.cpp

```
#include "../..//include/excecoes/ExcecaoArquivos.hpp"
```

Referência do Arquivo src/excecoes/ExcecaoBusca.cpp

```
#include "../..//include/excecoes/ExcecaoBusca.hpp"
```

Referência do Arquivo src/excecoes/ExcecaoCadastro.cpp

```
#include "../..//include/excecoes/ExcecaoCadastro.hpp"
```

Referência do Arquivo src/excecoes/ExcecaoLogin.cpp

```
#include "../..//include/excecoes/ExcecaoLogin.hpp"
```

Referência do Arquivo

src/excecoes/ExcecaoManipulacaoLivros.cpp

```
#include "../..//include/excecoes/ExcecaoManipulacaoLivros.hpp"
```

Referência do Arquivo src/gestao/Pessoa.cpp

```
#include "../..//include/gestao/Pessoa.hpp"
```

Referência do Arquivo src/gestao/Usuario.cpp

```
#include "../..//include/gestao/Usuario.hpp"
```


Referência do Arquivo src/login/Login.cpp

```
#include "../..//include/login/Login.hpp"  
#include "../..//include/excecoes/ExcecaoArquivos.hpp"  
#include "../..//include/excecoes/ExcecaoLogin.hpp"
```

Referência do Arquivo src/menu/Menu.cpp

```
#include "../..//include/menu/Menu.hpp"
```

Sumário

INDEX