

Trabalho 1 de Técnicas de Programação



Guilherme Braga e Gabriel Matheus
Versão 1.0
Terça, 30 de Abril de 2019

Índice dos Componentes

Lista de Classes

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

Apresentacao (Apresentação)	5
Cartao_de_credito (Cartão de Crédito)	7
Cidade (Cidade)	9
Classe_Evento (Classe de Evento)	10
Codigo_de_Apresentacao (Código de Apresentação)	11
Codigo_de_Evento (Código de Evento)	12
Codigo_de_Ingresso (Código de Ingresso)	13
Codigo_de_Seguranca_Cartao_Credito (Código de Segurança de Cartão Crédito)	14
CPF (CPF)	15
Data (Data)	16
Data_Validade_Cartao_Credito (Data de Validade do Cartão Crédito)	17
Disponibilidade (Disponibilidade)	18
Estados_Brasileiros (Estados Brasileiros)	19
Evento (Evento)	20
Faixa_Etaria (Faixa Etária)	22
Horario (Horário)	23
Ingresso (Ingresso)	24
Nome_de_Evento (Nome de Evento)	25
Numero_Cartao_Credito (Número de Cartão Crédito)	26
Numero_de_Sala (Número de sala)	27
Preco (Preço)	28
Senha (Senha)	29
Usuario (Usuário)	30

Índice dos Arquivos

Lista de Arquivos

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

dominios.cpp (Arquivo com a implementação dos métodos das classes)31
dominios.h (Arquivo com a declaração das classes de dominio do programa)32
entidades.cpp (Arquivo com a declaração das entidades do programa)34
entidades.h (Arquivo com a declaração das classes de entidade do programa)35
main.cpp (Arquivo com a implementação básica do programa, onde chamamos os testes)36

Classes

Referência da Classe Apresentacao

Apresentação.

```
#include <entidades.h>
```

Membros Públicos

- void **setApresentacao** (int novo_codigo, string novo_data, string novo_horario, float novo_preco, int novo_sala, int novo_disponibilidade) throw (invalid_argument)
- void **getApresentacao** (Codigo_de_Apresentacao *codigo_f, Data *data_f, Horario *horario_f, Preco *preco_f, Numero_de_Sala *sala_f, Disponibilidade *disponibilidade_f)

Descrição detalhada

Apresentação.

Autor:

Guilherme Braga
Gabriel Matheus

DESCRIÇÃO

Essa classe armazena um objeto de código de apresentação, de data, de horário, de preço e de disponibilidade.

FORMA

A atribuição dos objetos devem obedecer à suas formações respectivas no nível de domínio.

Funções membros

void Apresentacao::getApresentacao (Codigo_de_Apresentacao * *codigo_f*, Data * *data_f*, Horario * *horario_f*, Preco * *preco_f*, Numero_de_Sala * *sala_f*, Disponibilidade * *disponibilidade_f*)

Obtém os dados do Evento.

Retorna:

Código de apresentação, data, horário, preço e disponibilidade.

void Apresentacao::setApresentacao (int *novo_codigo*, string *novo_data*, string *novo_horario*, float *novo_preco*, int *novo_sala*, int *novo_disponibilidade*) throw (invalid_argument)

Atribui valor para os objetos que descrevem tal apresentação.

Parâmetros:

<i>int, string.</i>	
---------------------	--

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

- entidades.h
- entidades.cpp

Referência da Classe Cartao_de_credito

Cartão de Crédito.

```
#include <entidades.h>
```

Membros Públicos

- void **setCartao_de_credito** (long long int numero_f, int codigo_cartao_f, string data_f) throw (invalid_argument)
- void **getCartao_de_credito** (Numero_Cartao_Credito *numero_f, Codigo_de_Seguranca_Cartao_Credito *codigo_cartao_f, Data_Validade_Cartao_Credito *data_f)

Descrição detalhada

Cartão de Crédito.

Autor:

Guilherme Braga
Gabriel Matheus

DESCRIÇÃO

Essa classe armazena um objeto de número de cartão de crédito, um código de segurança de cartão de crédito e sua respectiva data de validade.

FORMA

A atribuição dos objetos devem obedecer à suas formações respectivas no nível de domínio.

Funções membros

```
void Cartao_de_credito::getCartao_de_credito (Numero_Cartao_Credito * numero_f,  
Codigo_de_Seguranca_Cartao_Credito * codigo_cartao_f,  
Data_Validade_Cartao_Credito * data_f)
```

Obtém os dados do Cartão de Crédito.

Retorna:

Número de cartão de crédito, código de segurança e sua validade.

```
void Cartao_de_credito::setCartao_de_credito (long long int numero_f, int  
codigo_cartao_f, string data_f) throw ( invalid_argument)
```

Atribui valor para o objeto de código de ingresso do ingresso propriamente dito.

Parâmetros:

<i>long</i>	long int, int e string.
-------------	-------------------------

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

- entidades.h
- entidades.cpp

Referência da Classe Cidade

Cidade.

```
#include <dominios.h>
```

Membros Públicos

- void **setCidade** (string) throw (invalid_argument)
 - string **getCidade** ()
-

Descrição detalhada

Cidade.

Autor:

Guilherme Braga
Gabriel Matheus

DESCRIÇÃO

Essa classe armazena um tipo char.

FORMA

Temos que checar se existem no máximo 15 caracteres e se a entrada obedece as especificações de um nome de cidade (não podemos ter espaços consecutivos, por exemplo).

Funções membros

string Cidade::getCidade () [inline]

Obtém o nome da **Cidade**.

Retorna:

string

void Cidade::setCidade (string *nova_cidade*) throw (invalid_argument)

Atribui valor para a cidade.

Parâmetros:

string	
--------	--

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

- dominios.h
- dominios.cpp

Referência da Classe Classe_Evento

Classe de **Evento**.

```
#include <dominios.h>
```

Membros Públicos

- void **setClasse_Evento** (int) throw (invalid_argument)
 - int **getClasse_Evento** () const
-

Descrição detalhada

Classe de **Evento**.

Autor:

Guilherme Braga
Gabriel Matheus

DESCRIÇÃO

Essa classe armazena um tipo int.

FORMA

O inteiro pode ser entre 1 e 4, representando cada número um tipo de evento diferente.

Funções membros

int Classe_Evento::getClasse_Evento () const [inline]

Obtém a Classe de **Evento**.

Retorna:

int.

void Classe_Evento::setClasse_Evento (int novo_classe_evento) throw (invalid_argument)

Atribui valor para a Classe de **Evento**.

Parâmetros:

int	
-----	--

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

- dominios.h
- dominios.cpp

Referência da Classe `Codigo_de_Apresentacao`

Código de Apresentação.

```
#include <dominios.h>
```

Membros Públicos

- `void setCodigo_de_Apresentacao (int) throw (invalid_argument)`
 - `int getCodigo_de_Apresentacao () const`
-

Descrição detalhada

Código de Apresentação.

Autor:

Guilherme Braga
Gabriel Matheus

DESCRIÇÃO

Essa classe armazena um inteiro que representa o Código de Apresentação.

FORMA

O inteiro deve apresentar 4 dígitos apenas.

Funções membros

`int Codigo_de_Apresentacao::getCodigo_de_Apresentacao () const [inline]`

Obtém o código de apresentação.

Retorna:

Um inteiro.

`void Codigo_de_Apresentacao::setCodigo_de_Apresentacao (int novo_codigo_de_apresentacao) throw (invalid_argument)`

Atribui valor para código de apresentação.

Parâmetros:

<i>int</i>	
------------	--

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

- `dominios.h`
- `dominios.cpp`

Referência da Classe `Codigo_de_Evento`

Código de **Evento**.

```
#include <dominios.h>
```

Membros Públicos

- `void setCodigo_de_Evento (int) throw (invalid_argument)`
 - `int getCodigo_de_Evento () const`
-

Descrição detalhada

Código de **Evento**.

Autor:

Guilherme Braga
Gabriel Matheus

DESCRIÇÃO

Essa classe armazena um inteiro que representa o Código de **Evento**

FORMA

O inteiro deve apresentar 3 dígitos apenas

Funções membros

`int Codigo_de_Evento::getCodigo_de_Evento () const [inline]`

Obtém o código de evento

Retorna:

Um inteiro.

`void Codigo_de_Evento::setCodigo_de_Evento (int novo_codigo_de_evento) throw (invalid_argument)`

Atribui valor para código de evento.

Parâmetros:

<i>int</i>	
------------	--

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

- `dominios.h`
- `dominios.cpp`

Referência da Classe `Codigo_de_Ingresso`

Código de **Ingresso**.

```
#include <dominios.h>
```

Membros Públicos

- `void setCodigo_de_Ingresso (int) throw (invalid_argument)`
 - `int getCodigo_de_Ingresso () const`
-

Descrição detalhada

Código de **Ingresso**.

Autor:

Guilherme Braga
Gabriel Matheus

DESCRIÇÃO

Essa classe armazena um inteiro que representa o Código de **Ingresso**.

FORMA

O inteiro deve apresentar 5 dígitos apenas.

Funções membros

`int Codigo_de_Ingresso::getCodigo_de_Ingresso () const [inline]`

Obtém o código de ingresso.

Retorna:

Um inteiro.

`void Codigo_de_Ingresso::setCodigo_de_Ingresso (int novo_codigo_de_ingresso) throw (invalid_argument)`

Atribui valor para código de ingresso.

Parâmetros:

<i>int</i>	
------------	--

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

- `dominios.h`
- `dominios.cpp`

Referência da Classe `Codigo_de_Seguranca_Cartao_Credito`

Código de Segurança de Cartão Crédito.

```
#include <dominios.h>
```

Membros Públicos

- `void setCodigo_de_Seguranca_Cartao_Credito (int) throw (invalid_argument)`
 - `int getCodigo_de_Seguranca_Cartao_Credito () const`
-

Descrição detalhada

Código de Segurança de Cartão Crédito.

Autor:

Guilherme Braga

Gabriel Matheus

DESCRIÇÃO

Essa classe armazena um tipo `int`.

FORMA

Temos que checar se o código de segurança apresenta 3 dígitos.

Funções membros

```
int Codigo_de_Seguranca_Cartao_Credito::getCodigo_de_Seguranca_Cartao_Credito  
() const [inline]
```

Obtém o código de segurança.

Retorna:

`int`.

void

```
Codigo_de_Seguranca_Cartao_Credito::setCodigo_de_Seguranca_Cartao_Credito (int  
novo_codigo_de_seguranca) throw ( invalid_argument)
```

Atribui valor para o código de segurança.

Parâmetros:

<code>int</code>	
------------------	--

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

- `dominios.h`
- `dominios.cpp`

Referência da Classe CPF

CPF.

```
#include <dominios.h>
```

Membros Públicos

- void **setCPF** (long long int) throw (invalid_argument)
 - long long int **getCPF** ()
-

Descrição detalhada

CPF.

Autor:

Guilherme Braga
Gabriel Matheus

DESCRIÇÃO

Essa classe armazena um tipo long long int.

FORMA

O CPF deve obedecer o algoritmo de validação de um CPF.

Funções membros

long long int CPF::getCPF () [inline]

Obtém o CPF.

Retorna:

long long int.

void CPF::setCPF (long long int novo_num_cpf) throw (invalid_argument)

Atribui valor para o CPF.

Parâmetros:

<i>long</i>	long int
-------------	----------

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

- dominios.h
- dominios.cpp

Referência da Classe Data

Data.

```
#include <dominios.h>
```

Membros Públicos

- void **setData** (string) throw (invalid_argument)
 - string **getData** ()
-

Descrição detalhada

Data.

Autor:

Guilherme Braga
Gabriel Matheus

DESCRIÇÃO

Essa classe armazena um tipo char.

FORMA

Temos que checar se existem no máximo 6 caracteres e se a entrada obedece o formato correto de data: DDMMAA.

Funções membros

string Data::getData () [inline]

Obtém a data.

Retorna:

string

void Data::setData (string nova_data) throw (invalid_argument)

Atribui valor para a data.

Parâmetros:

string	
--------	--

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

- dominios.h
- dominios.cpp

Referência da Classe Data_Validade_Cartao_Credito

Data de Validade do Cartão Crédito.

```
#include <dominios.h>
```

Membros Públicos

- void setData_Validade_Cartao_Credito (string) throw (invalid_argument)
 - string getData_Validade_Cartao_Credito ()
-

Descrição detalhada

Data de Validade do Cartão Crédito.

Autor:

Guilherme Braga

Gabriel Matheus

DESCRIÇÃO

Essa classe armazena um tipo char.

FORMA

Temos que checar se o número apresenta 5 caracteres.

Funções membros

string Data_Validade_Cartao_Credito::getData_Validade_Cartao_Credito () [inline]

Obtém a validade do cartão.

Retorna:

string

void Data_Validade_Cartao_Credito::setData_Validade_Cartao_Credito (string novo_data_validade) throw (invalid_argument)

Atribui valor para a validade do cartão.

Parâmetros:

string	
--------	--

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

- dominios.h
- dominios.cpp

Referência da Classe Disponibilidade

Disponibilidade.

```
#include <dominios.h>
```

Membros Públicos

- void **setDisponibilidade** (int) throw (invalid_argument)
 - int **getDisponibilidade** () const
-

Descrição detalhada

Disponibilidade.

Autor:

Guilherme Braga
Gabriel Matheus

DESCRIÇÃO

Essa classe armazena um tipo int.

FORMA

Temos que checar se o número apresenta 3 dígitos.

Funções membros

int Disponibilidade::getDisponibilidade () const [inline]

Obtém disponibilidade.

Retorna:

int

void Disponibilidade::setDisponibilidade (int novo_disponibilidade) throw (invalid_argument)

Atribui valor para a disponibilidade.

Parâmetros:

int	
-----	--

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

- dominios.h
- dominios.cpp

Referência da Classe Estados_Brasileiros

Estados Brasileiros.

```
#include <dominios.h>
```

Membros Públicos

- void **setEstado** (string) throw (invalid_argument)
 - string **getEstado** ()
-

Descrição detalhada

Estados Brasileiros.

Autor:

Guilherme Braga
Gabriel Matheus

DESCRIÇÃO

Essa classe armazena um tipo char.

FORMA

O char deve representar a sigla de um Estado Brasileiro existente.

Funções membros

string Estados_Brasileiros::getEstado () [inline]

Obtém o Estado Brasileiro.

Retorna:

string.

void Estados_Brasileiros::setEstado (string novo_estado_br) throw (invalid_argument)

Atribui valor para o Estado Brasileiro.

Parâmetros:

string	
--------	--

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

- dominios.h
- dominios.cpp

Referência da Classe Evento

Evento.

```
#include <entidades.h>
```

Membros Públicos

- void **setEvento** (int novo_codigo, string novo_nome, string novo_cidade, string novo_estado, int novo_classe, string novo_faixa) throw (invalid_argument)
- void **getEvento** (Codigo_de_Evento *codigo_f, Nome_de_Evento *nome_f, Cidade *cidade_f, Estados_Brasileiros *estado_f, Classe_Evento *classe_f, Faixa_Etaria *faixa_f)

Descrição detalhada

Evento.

Autor:

Guilherme Braga

Gabriel Matheus

DESCRIÇÃO

Essa classe armazena um objeto de código de evento, nome de evento, sua cidade, seu respectivo Estado Brasileiro, a classe do evento e sua faixa etária.

FORMA

A atribuição dos objetos devem obedecer à suas formações respectivas no nível de domínio.

Funções membros

void Evento::getEvento (Codigo_de_Evento * *codigo_f*, Nome_de_Evento * *nome_f*, Cidade * *cidade_f*, Estados_Brasileiros * *estado_f*, Classe_Evento * *classe_f*, Faixa_Etaria * *faixa_f*)

Obtém os dados do Evento.

Retorna:

Código de evento, nome de evento, cidade, Estado Brasileiro, classe de evento e faixa etária.

void Evento::setEvento (int *novo_codigo*, string *novo_nome*, string *novo_cidade*, string *novo_estado*, int *novo_classe*, string *novo_faixa*) throw (invalid_argument)

Atribui valor para os objetos que descrevem tal evento.

Parâmetros:

<i>int, string.</i>	
---------------------	--

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

- entidades.h
- entidades.cpp

Referência da Classe Faixa_Etaria

Faixa Etária.

```
#include <dominios.h>
```

Membros Públicos

- void **setFaixa_Etaria** (string) throw (invalid_argument)
- string **getFaixa_Etaria** ()

Descrição detalhada

Faixa Etária.

Autor:

Guilherme Braga
Gabriel Matheus

DESCRIÇÃO

Essa classe armazena um tipo char.

FORMA

O char deve representar uma faixa etária existente no Brasil (L, 10, 12, 14, 16 ou 18).

Funções membros

string Faixa_Etaria::getFaixa_Etaria () [inline]

Obtém a Faixa Etária.

Retorna:

string.

void Faixa_Etaria::setFaixa_Etaria (string *novo_faixa_etaria*) throw (invalid_argument)

Atribui valor para a Faixa Etária.

Parâmetros:

<i>string</i>	
---------------	--

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

- dominios.h
- dominios.cpp

Referência da Classe Horário

Horário.

```
#include <dominios.h>
```

Membros Públicos

- void **setHorario** (string) throw (invalid_argument)
 - string **getHorario** ()
-

Descrição detalhada

Horário.

Autor:

Guilherme Braga
Gabriel Matheus

DESCRIÇÃO

Essa classe armazena um char.

FORMA

O inteiro deve se apresentar na forma de HH:MM.

Funções membros

string Horário::getHorario () [inline]

Obtém o horário.

Retorna:

string.

void Horário::setHorario (string novo_horario) throw (invalid_argument)

Atribui valor para o horário.

Parâmetros:

string	
--------	--

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

- dominios.h
- dominios.cpp

Referência da Classe Ingresso

Ingresso.

```
#include <entidades.h>
```

Membros Públicos

- void **setIngresso** (int novo_codigo) throw (invalid_argument)
 - void **getIngresso** (Codigo_de_Ingresso *codif)
-

Descrição detalhada

Ingresso.

Autor:

Guilherme Braga
Gabriel Matheus

DESCRIÇÃO

Essa classe armazena um objeto de código de ingresso.

FORMA

A atribuição dos objetos devem obedecer à suas formações respectivas no nível de domínio.

Funções membros

void Ingresso::getIngresso (Codigo_de_Ingresso * codif)

Obtém os dados do **Ingresso**

Retorna:

Código de **Ingresso**.

void Ingresso::setIngresso (int novo_codigo) throw (invalid_argument)

Atribui valor para o objeto de código de ingresso do ingresso propriamente dito.

Parâmetros:

<i>int</i>	
------------	--

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

- entidades.h
- entidades.cpp

Referência da Classe Nome_de_Evento

Nome de **Evento**.

```
#include <dominios.h>
```

Membros Públicos

- void **setNome_de_Evento** (string) throw (invalid_argument)
 - string **getNome_de_Evento** ()
-

Descrição detalhada

Nome de **Evento**.

Autor:

Guilherme Braga
Gabriel Matheus

DESCRIÇÃO

Essa classe armazena um tipo char.

FORMA

Temos que checar se existem no máximo 20 caracteres.

Funções membros

string Nome_de_Evento::getNome_de_Evento () [inline]

Obtém nome de evento.

Retorna:

string

void Nome_de_Evento::setNome_de_Evento (string *novo_nome_evento*) throw (invalid_argument)

Atribui valor para o nome do evento.

Parâmetros:

string	
--------	--

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

- dominios.h
- dominios.cpp

Referência da Classe Numero_Cartao_Credito

Número de Cartão Crédito.

```
#include <dominios.h>
```

Membros Públicos

- void **setNumero_Cartao_Credito** (long long int) throw (invalid_argument)
- long long int **getNumero_Cartao_Credito** ()

Descrição detalhada

Número de Cartão Crédito.

Autor:

Guilherme Braga
Gabriel Matheus

DESCRIÇÃO

Essa classe armazena um tipo long long int.

FORMA

Temos que checar se o número apresenta 9 dígitos e se ele obedece o algoritmo de formação.

Funções membros

long long int Numero_Cartao_Credito::getNumero_Cartao_Credito () [inline]

Obtém o número do cartão.

Retorna:

long long int

void Numero_Cartao_Credito::setNumero_Cartao_Credito (long long int novo_num_cartao_credito) throw (invalid_argument)

Atribui valor para o número do cartão.

Parâmetros:

<i>long</i>	long int
-------------	----------

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

- dominios.h
- dominios.cpp

Referência da Classe Numero_de_Sala

Número de sala.

```
#include <dominios.h>
```

Membros Públicos

- void **setNumero_de_Sala** (int) throw (invalid_argument)
- int **getNumero_de_Sala** () const

Descrição detalhada

Número de sala.

Autor:

Guilherme Braga
Gabriel Matheus

DESCRIÇÃO

Essa classe armazena um inteiro.

FORMA

O inteiro deve ser maior que zero e menor ou igual a 10.

Funções membros

int Numero_de_Sala::getNumero_de_Sala () const [inline]

Obtém o número da sala.

Retorna:

inteiro.

void Numero_de_Sala::setNumero_de_Sala (int novo_numero_de_sala) throw (invalid_argument)

Atribui valor para o número de sala.

Parâmetros:

int	
-----	--

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

- dominios.h
- dominios.cpp

Referência da Classe Preço

Preço.
`#include <dominios.h>`

Membros Públicos

- `void setPreco (float) throw (invalid_argument)`
- `float getPreco () const`

Descrição detalhada

Preço.

Autor:

Guilherme Braga
Gabriel Matheus

DESCRIÇÃO

Essa classe armazena um float.

FORMA

O inteiro deve ser maior ou igual à zero zero e menor e menor ou igual a 1000.

Funções membros

`float Preço::getPreco () const [inline]`

Obtém o preço.

Retorna:

float.

`void Preço::setPreco (float novo_preco) throw (invalid_argument)`

Atribui valor para o preço.

Parâmetros:

<i>float</i>	
--------------	--

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

- `dominios.h`
- `dominios.cpp`

Referência da Classe Senha

Senha.

```
#include <dominios.h>
```

Membros Públicos

- void **setSenha** (string) throw (invalid_argument)
 - string **getSenha** () const
-

Descrição detalhada

Senha.

Autor:

Guilherme Braga
Gabriel Matheus

DESCRIÇÃO

Essa classe armazena um tipo char.

FORMA

A senha deve possuir uma letra maiúscula, uma minúscula e um dígito. Não podem existir caracteres repetidos.

Funções membros

string Senha::getSenha () const [inline]

Obtém a senha.

Retorna:

string.

void Senha::setSenha (string novo_senha) throw (invalid_argument)

Atribui valor para a senha.

Parâmetros:

string	
--------	--

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

- dominios.h
- dominios.cpp

Referência da Classe Usuario

Usuário.

```
#include <entidades.h>
```

Membros Públicos

- void **setUsuario** (long long int novo_num_cpf, string novo_senha) throw (invalid_argument)
- void **getUsuario** (CPF *cpf_f, Senha *senha_f)

Descrição detalhada

Usuário.

Autor:

Guilherme Braga
Gabriel Matheus

DESCRIÇÃO

Essa classe armazena um objeto de cpf e um de senha.

FORMA

A atribuição dos objetos devem obedecer à suas formações respectivas no nível de domínio.

Funções membros

void Usuario::getUsuario (CPF * *cpf_f*, Senha * *senha_f*)

Obtém os dados do Usuário

Retorna:

CPF e Senha.

void Usuario::setUsuario (long long int *novo_num_cpf*, string *novo_senha*) throw (invalid_argument)

Atribui valor para os objetos do usuário.

Parâmetros:

<i>long</i>	long int e string
-------------	-------------------

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

- entidades.h
- entidades.cpp

Arquivos

Referência do Arquivo dominios.cpp

Arquivo com a implementação dos métodos das classes.

```
#include <stdexcept>
#include <iostream>
#include <string>
#include <string.h>
#include <stdlib.h>
#include <ctype.h>
#include "dominios.h"
```

Descrição detalhada

Arquivo com a implementação dos métodos das classes.

Autor:

Gabriel e Guilherme

Referência do Arquivo dominios.h

Arquivo com a declaração das classes de domínio do programa.

```
#include <stdexcept>
#include <iostream>
#include <string>
#include <stdlib.h>
```

Componentes

- class **Codigo_de_Evento**
*Código de **Evento**.*
 - class **Codigo_de_Apresentacao**
Código de Apresentação.
 - class **Codigo_de_Ingresso**
*Código de **Ingresso**.*
 - class **Horario**
Horário.
 - class **Numero_de_Sala**
Número de sala.
 - class **Preco**
Preço.
 - class **Estados_Brasileiros**
Estados Brasileiros.
 - class **Classe_Evento**
*Classe de **Evento**.*
 - class **Faixa_Etaria**
Faixa Etária.
 - class **CPF**
***CPF**.*
 - class **Senha**
***Senha**.*
 - class **Codigo_de_Seguranca_Cartao_Credito**
Código de Segurança de Cartão Crédito.
 - class **Numero_Cartao_Credito**
Número de Cartão Crédito.
 - class **Data_Validade_Cartao_Credito**
***Data** de Validade do Cartão Crédito.*
 - class **Disponibilidade**
***Disponibilidade**.*
 - class **Nome_de_Evento**
*Nome de **Evento**.*
 - class **Data**
***Data**.*
 - class **Cidade**
***Cidade**.*
-

Descrição detalhada

Arquivo com a declaração das classes de domínio do programa.

Autor:

Gabriel e Guilherme

Referência do Arquivo entidades.cpp

Arquivo com a declaração das entidades do programa.

```
#include "dominios.h"  
#include <string>  
#include <iostream>
```

Descrição detalhada

Arquivo com a declaração das entidades do programa.

Autor:

Gabriel e Guilherme

Referência do Arquivo entidades.h

Arquivo com a declaração das classes de entidade do programa.

```
#include "dominios.h"  
#include <string>
```

Componentes

- class **Usuario**
Usuário.
 - class **Ingresso**
Ingresso.
 - class **Cartao_de_credito**
Cartão de Crédito.
 - class **Evento**
Evento.
 - class **Apresentacao**
Apresentação.
-

Descrição detalhada

Arquivo com a declaração das classes de entidade do programa.

Autor:

Gabriel e Guilherme

Referência do Arquivo main.cpp

Arquivo com a implementação básica do programa, onde chamamos os testes.

```
#include <stdexcept>
#include <iostream>
#include <string>
#include <stdlib.h>
#include <conio.h>
#include "dominios.h"
#include "dominios.cpp"
#include "entidades.h"
#include "entidades.cpp"
#include "testes.h"
#include "testes.cpp"
```

Funções

- `int main ()`

Descrição detalhada

Arquivo com a implementação básica do programa, onde chamamos os testes.

Autor:

Gabriel e Guilherme

