

## Projeto TP1

Generated by Doxygen 1.9.3



<b>1 Class Index</b>	<b>1</b>
1.1 Class List	1
<b>2 File Index</b>	<b>3</b>
2.1 File List	3
<b>3 Class Documentation</b>	<b>5</b>
3.1 Avaliacao Class Reference	5
3.1.1 Detailed Description	5
3.1.1.1 Declaracao da entidade Avaliacao que permite instanciar um codigo, nota e descricao.	5
3.1.2 Member Function Documentation	5
3.1.2.1 getCodigo()	5
3.1.2.2 getDescricao()	5
3.1.2.3 getNota()	5
3.1.2.4 setCodigo()	6
3.1.2.5 Implementacao Metodos inline set da entidade Avaliacao.	6
3.1.2.6 setDescricao()	7
3.1.2.7 setNota()	7
3.2 Cidade Class Reference	7
3.2.1 Detailed Description	7
3.2.1.1 Dominio Cidade que permite instanciar uma cidade.	7
3.2.2 Constructor & Destructor Documentation	7
3.2.2.1 Cidade()	7
3.2.3 Member Function Documentation	8
3.2.3.1 getCidade()	8
3.2.3.2 setCidade()	8
3.3 Codigo Class Reference	8
3.3.1 Detailed Description	8
3.3.1.1 Dominio Codigo que permite instanciar um codigo.	8
3.3.2 Constructor & Destructor Documentation	8
3.3.2.1 Codigo()	9
3.3.3 Member Function Documentation	9
3.3.3.1 getCodigo()	9
3.3.3.2 setCodigo()	9
3.4 Data Class Reference	9
3.4.1 Detailed Description	10
3.4.1.1 Dominio Data que permite instanciar uma data.	10
3.4.2 Constructor & Destructor Documentation	10
3.4.2.1 Data()	10
3.4.3 Member Function Documentation	10
3.4.3.1 getData()	10
3.4.3.2 setData()	10
3.5 Descricao Class Reference	10

3.5.1 Detailed Description . . . . .	11
3.5.1.1 Dominio Descricao que permite instanciar uma duracao. . . . .	11
3.5.2 Constructor & Destructor Documentation . . . . .	11
3.5.2.1 Descricao() . . . . .	11
3.5.3 Member Function Documentation . . . . .	11
3.5.3.1 getDescricao() . . . . .	11
3.5.3.2 setDescricao() . . . . .	11
3.6 Duracao Class Reference . . . . .	11
3.6.1 Detailed Description . . . . .	12
3.6.1.1 Dominio Duracao que permite instanciar uma duracao. . . . .	12
3.6.2 Constructor & Destructor Documentation . . . . .	12
3.6.2.1 Duracao() . . . . .	12
3.6.3 Member Function Documentation . . . . .	12
3.6.3.1 getDuracao() . . . . .	12
3.6.3.2 setDuracao() . . . . .	12
3.7 Email Class Reference . . . . .	12
3.7.1 Detailed Description . . . . .	13
3.7.1.1 Dominio Email que permite instanciar um email. . . . .	13
3.7.2 Constructor & Destructor Documentation . . . . .	13
3.7.2.1 Email() . . . . .	13
3.7.3 Member Function Documentation . . . . .	13
3.7.3.1 getEmail() . . . . .	13
3.7.3.2 setEmail() . . . . .	13
3.8 Endereco Class Reference . . . . .	13
3.8.1 Detailed Description . . . . .	14
3.8.1.1 Dominio Endereco que permite instanciar um endereco. . . . .	14
3.8.2 Constructor & Destructor Documentation . . . . .	14
3.8.2.1 Endereco() . . . . .	14
3.8.3 Member Function Documentation . . . . .	14
3.8.3.1 getEndereco() . . . . .	14
3.8.3.2 setEndereco() . . . . .	14
3.9 Excursao Class Reference . . . . .	15
3.9.1 Detailed Description . . . . .	15
3.9.1.1 Declaracao da entidade Excursao que permite instanciar um codigo, titulo e nota. . . . .	15
3.9.2 Member Function Documentation . . . . .	15
3.9.2.1 getCidade() . . . . .	15
3.9.2.2 getCodigo() . . . . .	15
3.9.2.3 getDescricao() . . . . .	15
3.9.2.4 getDuracao() . . . . .	15
3.9.2.5 getEndereco() . . . . .	15
3.9.2.6 getNota() . . . . .	15
3.9.2.7 getTitulo() . . . . .	16

3.9.2.8 setCidade()	16
3.9.2.9 setCodigo()	16
3.9.2.10 setDescricao()	16
3.9.2.11 setDuracao()	16
3.9.2.12 setEndereco()	16
3.9.2.13 setNota()	17
3.9.2.14 setTitulo()	17
3.10 Horario Class Reference	17
3.10.1 Detailed Description	17
3.10.1.1 Dominio Horario que permite instanciar um horario.	17
3.10.2 Constructor & Destructor Documentation	17
3.10.2.1 Horario()	17
3.10.3 Member Function Documentation	18
3.10.3.1 getHorario()	18
3.10.3.2 setHorario()	18
3.11 Idioma Class Reference	18
3.11.1 Detailed Description	18
3.11.1.1 Dominio Idioma que permite instanciar um idioma.	18
3.11.2 Constructor & Destructor Documentation	18
3.11.2.1 Idioma()	19
3.11.3 Member Function Documentation	19
3.11.3.1 getIdioma()	19
3.11.3.2 setIdioma()	19
3.12 Nome Class Reference	19
3.12.1 Detailed Description	20
3.12.1.1 Dominio Nome que permite instanciar um nome.	20
3.12.2 Constructor & Destructor Documentation	20
3.12.2.1 Nome()	20
3.12.3 Member Function Documentation	20
3.12.3.1 getNome()	20
3.12.3.2 setNome()	20
3.13 Nota Class Reference	20
3.13.1 Detailed Description	21
3.13.1.1 Dominio Nota que permite instanciar uma nota.	21
3.13.2 Constructor & Destructor Documentation	21
3.13.2.1 Nota()	21
3.13.3 Member Function Documentation	21
3.13.3.1 getNota()	21
3.13.3.2 setNota()	21
3.14 Senha Class Reference	21
3.14.1 Detailed Description	22
3.14.1.1 Dominio Senha que permite instanciar uma senha.	22

3.14.2 Constructor & Destructor Documentation	22
3.14.2.1 Senha()	22
3.14.3 Member Function Documentation	22
3.14.3.1 getSenha()	22
3.14.3.2 setSenha()	22
3.15 Sessao Class Reference	22
3.15.1 Detailed Description	23
3.15.1.1 Declaracao da entidade Sessao que permite instancia um codigo, data, horario e idioma.	23
3.15.2 Member Function Documentation	23
3.15.2.1 getCodigo()	23
3.15.2.2 getData()	23
3.15.2.3 getHorario()	23
3.15.2.4 getIdioma()	23
3.15.2.5 setCodigo()	23
3.15.2.6 setData()	23
3.15.2.7 setHorario()	23
3.15.2.8 setIdioma()	24
3.16 Titulo Class Reference	24
3.16.1 Detailed Description	24
3.16.1.1 Dominio Titulo que permite instanciar um titulo.	24
3.16.2 Constructor & Destructor Documentation	24
3.16.2.1 Titulo()	24
3.16.3 Member Function Documentation	24
3.16.3.1 getTitulo()	25
3.16.3.2 setTitulo()	25
3.17 TUAvaliacao Class Reference	25
3.18 TUCidade Class Reference	25
3.19 TUCodigo Class Reference	26
3.20 TUData Class Reference	26
3.21 TUDescricao Class Reference	26
3.22 TUDuracao Class Reference	27
3.23 TUEmail Class Reference	27
3.24 TUEndereco Class Reference	27
3.25 TUExcursao Class Reference	28
3.26 TUHorario Class Reference	28
3.27 TUIdioma Class Reference	28
3.28 TUNome Class Reference	28
3.29 TUNota Class Reference	29
3.30 TUSenha Class Reference	29
3.31 TUSessao Class Reference	29
3.32 TUTitulo Class Reference	30

3.33 TUUsuario Class Reference . . . . .	30
3.34 Usuario Class Reference . . . . .	30
3.34.1 Detailed Description . . . . .	31
3.34.1.1 Declaracao da entidade Usuario que permite instanciar um nome, email e senha. . . . .	31
3.34.2 Member Function Documentation . . . . .	31
3.34.2.1 getEmail() . . . . .	31
3.34.2.2 getNome() . . . . .	31
3.34.2.3 getSenha() . . . . .	31
3.34.2.4 setEmail() . . . . .	31
3.34.2.5 setNome() . . . . .	31
3.34.2.6 setSenha() . . . . .	31
<b>4 File Documentation</b> . . . . .	<b>33</b>
4.1 dominios.h . . . . .	33
4.2 entidades.h . . . . .	36
4.3 testes.h . . . . .	39
<b>Index</b> . . . . .	<b>45</b>





# Chapter 1

## Class Index

### 1.1 Class List

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

Avaliacao	5
Cidade	7
Codigo	8
Data	9
Descricao	10
Duracao	11
Email	12
Endereco	13
Excursao	15
Horario	17
Idioma	18
Nome	19
Nota	20
Senha	21
Sessao	22
Titulo	24
TUAvaliacao	25
TUCidade	25
TUCodigo	26
TUData	26
TUDescricao	26
TUDuracao	27
TUEmail	27
TUEndereco	27
TUExcursao	28
TUHorario	28
TUIdioma	28
TUNome	28
TUNota	29
TUSenha	29
TUSessao	29
TUTitulo	30
TUUsuario	30
Usuario	30



## Chapter 2

# File Index

### 2.1 File List

Here is a list of all documented files with brief descriptions:

C:/Users/felip/Downloads/T1-TP1-200060295-202021767-202021800/ <a href="#">dominios.h</a> . . . . .	33
C:/Users/felip/Downloads/T1-TP1-200060295-202021767-202021800/ <a href="#">entidades.h</a> . . . . .	36
C:/Users/felip/Downloads/T1-TP1-200060295-202021767-202021800/ <a href="#">testes.h</a> . . . . .	39



## Chapter 3

# Class Documentation

### 3.1 Avaliacao Class Reference

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

#### Public Member Functions

- void [setCodigo](#) (const [Codigo](#) &)
- void [setNota](#) (const [Nota](#) &)
- void [setDescricao](#) (const [Descricao](#) &)
- [Codigo](#) [getCodigo](#) () const
- [Nota](#) [getNota](#) () const
- [Descricao](#) [getDescricao](#) () const

#### 3.1.1 Detailed Description

3.1.1.1 Declaracao da entidade Avaliacao que permite instanciar um codigo, nota e descricao.

---

#### 3.1.2 Member Function Documentation

##### 3.1.2.1 [getCodigo\(\)](#)

[Codigo](#) Avaliacao::getCodigo ( ) const [inline]  
Metodo que recupera o codigo return string codigo

##### 3.1.2.2 [getDescricao\(\)](#)

[Descricao](#) Avaliacao::getDescricao ( ) const [inline]  
Metodo que recupera a descricao return string descricao

##### 3.1.2.3 [getNota\(\)](#)

[Nota](#) Avaliacao::getNota ( ) const [inline]  
Metodo que recupera a nota return string nota

#### 3.1.2.4 setCodigo()

```
void Avaliacao::setCodigo (
    const Codigo & codigo ) [inline]
```

---

#### 3.1.2.5 Implementacao Metodos inline set da entidade Avaliacao.

Metodo que seta o codigo

## Parameters

<code>string</code>	codigo return void
---------------------	--------------------

**3.1.2.6 setDescricao()**

```
void Avaliacao::setDescricao (
    const Descricao & descricao ) [inline]
```

Método que seta a descricao

## Parameters

<code>string</code>	descricao return void
---------------------	-----------------------

**3.1.2.7 setNota()**

```
void Avaliacao::setNota (
    const Nota & nota ) [inline]
```

Método que seta a nota

## Parameters

<code>string</code>	nota return void
---------------------	------------------

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

- C:/Users/felip/Downloads/T1-TP1-200060295-202021767-202021800/entidades.h

**3.2 Cidade Class Reference**

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

**Public Member Functions**

- **Cidade** ()  
*Construtor [Cidade](#) vazio.*
- [Cidade](#) (string)
- void [setCidade](#) (string)
- string [getCidade](#) () const

**3.2.1 Detailed Description**


---

**3.2.1.1** **Dominio Cidade** que permite instanciar uma cidade.

**3.2.2 Constructor & Destructor Documentation****3.2.2.1 Cidade()**

```
Cidade::Cidade (
    string cidade )
```

Construtor [Cidade](#) que recebe uma string para inicializar um objeto

## Parameters

<code>string</code>	
---------------------	--

### 3.2.3 Member Function Documentation

#### 3.2.3.1 getCidade()

```
string Cidade::getCidade ( ) const [inline]
```

Funcao getCidade, que instancia o objeto com uma string

## Parameters

<code>string</code>	return string cidade
---------------------	----------------------

#### 3.2.3.2 setCidade()

```
void Cidade::setCidade (
    string cidade )
```

Funcao setCidade, que instancia o objeto com uma string

## Parameters

<code>string</code>	return void
---------------------	-------------

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

- C:/Users/felip/Downloads/T1-TP1-200060295-202021767-202021800/dominios.h
- C:/Users/felip/Downloads/T1-TP1-200060295-202021767-202021800/dominios.cpp

## 3.3Codigo Class Reference

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

### Public Member Functions

- **Codigo** ()  
*Construtor [Codigo](#) vazio.*
- [Codigo](#) (string)
- void [setCodigo](#) (string)
- string [getCodigo](#) () const

#### 3.3.1 Detailed Description

---

**3.3.1.1** Dominio Codigo que permite instanciar um codigo.

#### 3.3.2 Constructor & Destructor Documentation



### 3.3.2.1 `Codigo()`

```
Codigo::Codigo (
    string codigo )
```

Construtor `Codigo` que recebe uma string para inicializar um objeto

#### Parameters

<code>string</code>	
---------------------	--

## 3.3.3 Member Function Documentation

### 3.3.3.1 `getCodigo()`

```
string Codigo::getCodigo ( ) const [inline]
```

Funcao `getCodigo`, que instancia o objeto com uma string

#### Parameters

<code>string</code>	return string codigo
---------------------	----------------------

### 3.3.3.2 `setCodigo()`

```
void Codigo::setCodigo (
    string codigo )
```

Funcao `setCodigo`, que instancia o objeto com uma string

#### Parameters

<code>string</code>	return void
---------------------	-------------

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

- C:/Users/felip/Downloads/T1-TP1-200060295-202021767-202021800/dominios.h
- C:/Users/felip/Downloads/T1-TP1-200060295-202021767-202021800/dominios.cpp

## 3.4 Data Class Reference

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

### Public Member Functions

- `Data ()`  
*Construtor `Data` vazio.*
- `Data (string)`
- `void setData (string)`
- `string getData () const`

### 3.4.1 Detailed Description

#### 3.4.1.1 Dominio Data que permite instanciar uma data.

### 3.4.2 Constructor & Destructor Documentation

#### 3.4.2.1 Data()

```
Data::Data (
    string data )
```

Construtor [Data](#) que recebe uma string para inicializar um objeto

##### Parameters

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

### 3.4.3 Member Function Documentation

#### 3.4.3.1 getData()

```
string Data::getData ( ) const [inline]
```

Funcao getData

##### Returns

string data

#### 3.4.3.2 setData()

```
void Data::setData (
    string data )
```

Funcao setData, que instancia o objeto com uma string

##### Parameters

<i>string</i>	return void
---------------	-------------

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

- C:/Users/felip/Downloads/T1-TP1-200060295-202021767-202021800/dominios.h
- C:/Users/felip/Downloads/T1-TP1-200060295-202021767-202021800/dominios.cpp

## 3.5 Descricao Class Reference

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

### Public Member Functions

- **Descricao ()**  
*Construtor [Descricao](#) vazio.*
- [Descricao](#) (string)
- void [setDescricao](#) (string)
- string [getDescricao](#) () const

### 3.5.1 Detailed Description

#### 3.5.1.1 Dominio Descricao que permite instanciar uma duracao.

### 3.5.2 Constructor & Destructor Documentation

#### 3.5.2.1 Descricao()

```
Descricao::Descricao (
    string descricao )
```

Construtor [Descricao](#) que recebe uma string para inicializar um objeto

##### Parameters

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

### 3.5.3 Member Function Documentation

#### 3.5.3.1 getDescricao()

```
string Descricao::getDescricao ( ) const [inline]
```

Funcao getDescricao, que instancia o objeto com uma string

##### Parameters

<i>string</i>	return string descricao
---------------	-------------------------

#### 3.5.3.2 setDescricao()

```
void Descricao::setDescricao (
    string descricao )
```

Funcao setDescricao, que instancia o objeto com uma string

##### Parameters

<i>string</i>	return void
---------------	-------------

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

- C:/Users/felip/Downloads/T1-TP1-200060295-202021767-202021800/dominios.h
- C:/Users/felip/Downloads/T1-TP1-200060295-202021767-202021800/dominios.cpp

## 3.6 Duracao Class Reference

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

### Public Member Functions

- [Duracao](#) ()  
*Construtor [Duracao](#) vazio.*
- [Duracao](#) (string)

- void `setDuracao` (string)
- string `getDuracao` () const

### 3.6.1 Detailed Description

3.6.1.1 Dominio Duracao que permite instanciar uma duracao.

### 3.6.2 Constructor & Destructor Documentation

#### 3.6.2.1 Duracao()

```
Duracao::Duracao (
    string duracao )
```

Construtor `Duracao` que recebe uma string para inicializar um objeto

##### Parameters

<code>string</code>	
---------------------	--

### 3.6.3 Member Function Documentation

#### 3.6.3.1 getDuracao()

```
string Duracao::getDuracao ( ) const [inline]
```

Funcao `getDuracao`, que instancia o objeto com uma string

##### Parameters

<code>string</code>	return string duracao
---------------------	-----------------------

#### 3.6.3.2 setDuracao()

```
void Duracao::setDuracao (
    string duracao )
```

Funcao `setDuracao`, que instancia o objeto com uma string

##### Parameters

<code>string</code>	return void
---------------------	-------------

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

- C:/Users/felip/Downloads/T1-TP1-200060295-202021767-202021800/dominios.h
- C:/Users/felip/Downloads/T1-TP1-200060295-202021767-202021800/dominios.cpp

## 3.7 Email Class Reference

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

### Public Member Functions

- `Email` ()

Construtor [Email](#) vazio.

- [Email](#) (string)
- void [setEmail](#) (string)
- string [getEmail](#) () const

### 3.7.1 Detailed Description

3.7.1.1 Dominio Email que permite instanciar um email.

### 3.7.2 Constructor & Destructor Documentation

#### 3.7.2.1 Email()

```
Email::Email (
    string email )
```

Construtor [Email](#) que recebe string para inicializar um objeto.

##### Parameters

string	
--------	--

### 3.7.3 Member Function Documentation

#### 3.7.3.1 getEmail()

```
string Email::getEmail ( ) const [inline]
Funcao getEmail
```

##### Returns

string email

#### 3.7.3.2 setEmail()

```
void Email::setEmail (
    string email )
```

Funcao setEmail, que instancia o objeto com uma string

##### Parameters

string	return void
--------	-------------

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

- C:/Users/felip/Downloads/T1-TP1-200060295-202021767-202021800/dominios.h
- C:/Users/felip/Downloads/T1-TP1-200060295-202021767-202021800/dominios.cpp

## 3.8 Endereco Class Reference

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

## Public Member Functions

- **Endereco** ()  
*Construtor [Endereco](#) vazio.*
- [Endereco](#) (string)
- void [setEndereco](#) (string)
- string [getEndereco](#) () const

### 3.8.1 Detailed Description

**3.8.1.1** **Dominio Endereco que permite instanciar um endereco.**

### 3.8.2 Constructor & Destructor Documentation

#### 3.8.2.1 Endereco()

```
Endereco::Endereco (
    string endereco )
```

Construtor [Endereco](#) que recebe uma string para inicializar um objeto

##### Parameters

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

### 3.8.3 Member Function Documentation

#### 3.8.3.1 getEndereco()

```
string Endereco::getEndereco ( ) const [inline]
```

Funcao getEndereco, que instancia o objeto com uma string

##### Parameters

<i>string</i>	return string endereco
---------------	------------------------

#### 3.8.3.2 setEndereco()

```
void Endereco::setEndereco (
    string endereco )
```

Funcao setEndereco, que instancia o objeto com uma string

##### Parameters

<i>string</i>	return void
---------------	-------------

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

- C:/Users/felip/Downloads/T1-TP1-200060295-202021767-202021800/dominios.h
- C:/Users/felip/Downloads/T1-TP1-200060295-202021767-202021800/dominios.cpp

## 3.9 Excursao Class Reference

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

### Public Member Functions

- void [setCodigo](#) (const [Codigo](#) &)
- void [setTitulo](#) (const [Titulo](#) &)
- void [setNota](#) (const [Nota](#) &)
- void [setCidade](#) (const [Cidade](#) &)
- void [setDuracao](#) (const [Duracao](#) &)
- void [setDescricao](#) (const [Descricao](#) &)
- void [setEndereco](#) (const [Endereco](#) &)
- [Codigo](#) [getCodigo](#) () const
- [Titulo](#) [getTitulo](#) () const
- [Nota](#) [getNota](#) () const
- [Cidade](#) [getCidade](#) () const
- [Duracao](#) [getDuracao](#) () const
- [Descricao](#) [getDescricao](#) () const
- [Endereco](#) [getEndereco](#) () const

### 3.9.1 Detailed Description

---

**3.9.1.1** Declaracao da entidade Excursao que permite instanciar um codigo, titulo e nota.

### 3.9.2 Member Function Documentation

#### 3.9.2.1 [getCidade\(\)](#)

[Cidade](#) Excursao::getCidade ( ) const [inline]

Metodo que recupera a cidade return string cidade

#### 3.9.2.2 [getCodigo\(\)](#)

[Codigo](#) Excursao::getCodigo ( ) const [inline]

Metodo que recupera o codigo return string codigo

#### 3.9.2.3 [getDescricao\(\)](#)

[Descricao](#) Excursao::getDescricao ( ) const [inline]

Metodo que recupera a descricao return string descricao

#### 3.9.2.4 [getDuracao\(\)](#)

[Duracao](#) Excursao::getDuracao ( ) const [inline]

Metodo que recupera a duracao return string duracao

#### 3.9.2.5 [getEndereco\(\)](#)

[Endereco](#) Excursao::getEndereco ( ) const [inline]

Metodo que recupera o endereco return string endereco

#### 3.9.2.6 [getNota\(\)](#)

[Nota](#) Excursao::getNota ( ) const [inline]

Metodo que recupera a nota return string nota

### 3.9.2.7 getTitulo()

```
Titulo Excursao::getTitulo ( ) const [inline]
```

Metodo que recupera o titulo return string titulo

### 3.9.2.8 setCidade()

```
void Excursao::setCidade (
    const Cidade & cidade ) [inline]
```

Metodo que seta a cidade

#### Parameters

<i>string</i>	cidade return void
---------------	--------------------

### 3.9.2.9 setCodigo()

```
void Excursao::setCodigo (
    const Codigo & codigo ) [inline]
```

Metodo que seta o codigo

#### Parameters

<i>string</i>	codigo return void
---------------	--------------------

### 3.9.2.10 setDescricao()

```
void Excursao::setDescricao (
    const Descricao & descricao ) [inline]
```

Metodo que seta a descricao

#### Parameters

<i>string</i>	descricao return void
---------------	-----------------------

### 3.9.2.11 setDuracao()

```
void Excursao::setDuracao (
    const Duracao & duracao ) [inline]
```

Metodo que seta a duracao

#### Parameters

<i>string</i>	duracao return void
---------------	---------------------

### 3.9.2.12 setEndereco()

```
void Excursao::setEndereco (
    const Endereco & endereco ) [inline]
```

Metodo que seta o endereco



## Parameters

<code>string</code>	endereco return void
---------------------	----------------------

**3.9.2.13 setNota()**

```
void Excursao::setNota (
    const Nota & nota ) [inline]
```

Método que seta a nota

## Parameters

<code>string</code>	nota return void
---------------------	------------------

**3.9.2.14 setTitulo()**

```
void Excursao::setTitulo (
    const Titulo & titulo ) [inline]
```

Método que seta o título

## Parameters

<code>string</code>	titulo return void
---------------------	--------------------

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

- C:/Users/felip/Downloads/T1-TP1-200060295-202021767-202021800/entidades.h

## 3.10 Horário Class Reference

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

### Public Member Functions

- **Horario ()**  
*Construtor [Horario](#) vazio.*
- [Horario](#) (string)
- void [setHorario](#) (string)
- string [getHorario](#) () const

### 3.10.1 Detailed Description

---

**3.10.1.1** Dominio Horário que permite instanciar um horário.

### 3.10.2 Constructor & Destructor Documentation

**3.10.2.1 Horario()**

```
Horario::Horario (
    string horario )
```

Construtor [Horario](#) que recebe uma string para inicializar um objeto

## Parameters

<code>string</code>	
---------------------	--

### 3.10.3 Member Function Documentation

#### 3.10.3.1 getHorario()

```
string Horario::getHorario ( ) const [inline]
```

Funcao getHorario, que instancia o objeto com uma string

## Parameters

<code>string</code>	return string horario
---------------------	-----------------------

#### 3.10.3.2 setHorario()

```
void Horario::setHorario (
    string horario )
```

Funcao setHorario, que instancia o objeto com uma string

## Parameters

<code>string</code>	return void
---------------------	-------------

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

- C:/Users/felip/Downloads/T1-TP1-200060295-202021767-202021800/dominios.h
- C:/Users/felip/Downloads/T1-TP1-200060295-202021767-202021800/dominios.cpp

## 3.11 Idioma Class Reference

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

### Public Member Functions

- **Idioma** ()  
*Construtor [Idioma](#) vazio.*
- [Idioma](#) (string)
- void [setIdioma](#) (string)
- string [getIdioma](#) () const

#### 3.11.1 Detailed Description

---

**3.11.1.1** **Domínio Idioma** que permite instanciar um idioma.

#### 3.11.2 Constructor & Destructor Documentation

### 3.11.2.1 Idioma()

```
Idioma::Idioma (
    string idioma )
```

Construtor [Idioma](#) que recebe uma string para inicializar um objeto

#### Parameters

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

## 3.11.3 Member Function Documentation

### 3.11.3.1 getIdioma()

```
string Idioma::getIdioma ( ) const [inline]
```

Funcao getIdioma, que instancia o objeto com uma string

#### Parameters

<i>string</i>	return string idioma
---------------	----------------------

### 3.11.3.2 setIdioma()

```
void Idioma::setIdioma (
    string idioma )
```

Funcao setIdioma, que instancia o objeto com uma string

#### Parameters

<i>string</i>	return void
---------------	-------------

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

- C:/Users/felip/Downloads/T1-TP1-200060295-202021767-202021800/dominios.h
- C:/Users/felip/Downloads/T1-TP1-200060295-202021767-202021800/dominios.cpp

## 3.12 Nome Class Reference

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

### Public Member Functions

- **Nome ()**  
*Construtor [Nome](#) vazio.*
- [Nome](#) (string)
- void [setName](#) (string)
- string [getName](#) () const

### 3.12.1 Detailed Description

3.12.1.1 **Dominio** Nome que permite instanciar um nome.

### 3.12.2 Constructor & Destructor Documentation

#### 3.12.2.1 Nome()

```
Nome::Nome (
    string nome )
```

Construtor [Nome](#) que recebe uma string para inicializar um objeto

##### Parameters

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

### 3.12.3 Member Function Documentation

#### 3.12.3.1 getName()

```
string Nome::getName ( ) const [inline]
```

Funcao getName

##### Returns

string nome

#### 3.12.3.2 setName()

```
void Nome::setName (
    string nome )
```

Funcao setNota, que instancia o objeto com uma string

##### Parameters

<i>string</i>	return void
---------------	-------------

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

- C:/Users/felip/Downloads/T1-TP1-200060295-202021767-202021800/dominios.h
- C:/Users/felip/Downloads/T1-TP1-200060295-202021767-202021800/dominios.cpp

## 3.13 Nota Class Reference

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

### Public Member Functions

- **Nota** ()  
*Construtor [Nota](#) vazio.*
- [Nota](#) (int)
- void [setNota](#) (int)
- int [getNota](#) () const

### 3.13.1 Detailed Description

3.13.1.1 Dominio Nota que permite instanciar uma nota.

### 3.13.2 Constructor & Destructor Documentation

#### 3.13.2.1 Nota()

```
Nota::Nota (
    int nota )
```

Construtor [Nota](#) que recebe int para inicializar um objeto

##### Parameters

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

### 3.13.3 Member Function Documentation

#### 3.13.3.1 getNota()

```
int Nota::getNota ( ) const [inline]
```

Funcao getNota

##### Returns

int nota

#### 3.13.3.2 setNota()

```
void Nota::setNota (
    int nota )
```

Funcao setNota, que instancia o objeto com um int

##### Parameters

<i>int</i>	return void
------------	-------------

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

- C:/Users/felip/Downloads/T1-TP1-200060295-202021767-202021800/dominios.h
- C:/Users/felip/Downloads/T1-TP1-200060295-202021767-202021800/dominios.cpp

## 3.14 Senha Class Reference

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

### Public Member Functions

- [Senha](#) ()  
Construtor [Senha](#) vazio.
- [Senha](#) (string)
- void [setSenha](#) (string)
- string [getSenha](#) () const

### 3.14.1 Detailed Description

#### 3.14.1.1 Dominio Senha que permite instanciar uma senha.

### 3.14.2 Constructor & Destructor Documentation

#### 3.14.2.1 Senha()

```
Senha::Senha (
    string senha )
```

Construtor [Senha](#) que recebe uma string para inicializar um objeto

##### Parameters

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

### 3.14.3 Member Function Documentation

#### 3.14.3.1 getSenha()

```
string Senha::getSenha ( ) const [inline]
```

Funcao getSenha

##### Returns

string senha

#### 3.14.3.2 setSenha()

```
void Senha::setSenha (
    string senha )
```

Funcao setSenha, que instancia o objeto com uma string

##### Parameters

<i>string</i>	return void
---------------	-------------

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

- C:/Users/felip/Downloads/T1-TP1-200060295-202021767-202021800/dominios.h
- C:/Users/felip/Downloads/T1-TP1-200060295-202021767-202021800/dominios.cpp

## 3.15 Sessao Class Reference

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

### Public Member Functions

- void [setCodigo](#) (const [Codigo](#) &)
- void [setData](#) (const [Data](#) &)
- void [setHorario](#) (const [Horario](#) &)
- void [setIdioma](#) (const [Idioma](#) &)
- [Codigo](#) [getCodigo](#) () const

- [Data](#) `getData ()` const
- [Horario](#) `getHorario ()` const
- [Idioma](#) `getIdioma ()` const

### 3.15.1 Detailed Description

**3.15.1.1** Declaracao da entidade Sessao que permite instancia um codigo, data, horario e idioma.

### 3.15.2 Member Function Documentation

#### 3.15.2.1 getCodigo()

```
Codigo Sessao::getCodigo ( ) const [inline]
```

Metodo que recupera o codigo return string codigo

#### 3.15.2.2 getData()

```
Data Sessao::getData ( ) const [inline]
```

Metodo que recupera a data return string data

#### 3.15.2.3 getHorario()

```
Horario Sessao::getHorario ( ) const [inline]
```

Metodo que recupera o horario return string horario

#### 3.15.2.4 getIdioma()

```
Idioma Sessao::getIdioma ( ) const [inline]
```

Metodo que recupera o idioma return string idioma

#### 3.15.2.5 setCodigo()

```
void Sessao::setCodigo (
    const Codigo & codigo ) [inline]
```

Metodo que seta o codigo

##### Parameters

<i>string</i>	codigo return void
---------------	--------------------

#### 3.15.2.6 setData()

```
void Sessao::setData (
    const Data & data ) [inline]
```

Metodo que seta a data

##### Parameters

<i>string</i>	data return void
---------------	------------------

#### 3.15.2.7 setHorario()

```
void Sessao::setHorario (
    const Horario & horario ) [inline]
```

Método que seta o horário

#### Parameters

<i>string</i>	horario return void
---------------	---------------------

### 3.15.2.8 setIdioma()

```
void Sessao::setIdioma (
    const Idioma & idioma ) [inline]
```

Método que seta o idioma

#### Parameters

<i>string</i>	idioma return void
---------------	--------------------

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

- C:/Users/felip/Downloads/T1-TP1-200060295-202021767-202021800/entidades.h

## 3.16 Titulo Class Reference

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

### Public Member Functions

- **Titulo ()**  
*Construtor [Titulo](#) vazio.*
- [Titulo](#) (string)
- void [setTitulo](#) (string)
- string [getTitulo](#) () const

### 3.16.1 Detailed Description

---

**3.16.1.1** **Dominio Titulo que permite instanciar um titulo.**

### 3.16.2 Constructor & Destructor Documentation

#### 3.16.2.1 Titulo()

```
Titulo::Titulo (
    string titulo )
```

Construtor [Titulo](#) que recebe uma string para inicializar um objeto

#### Parameters

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

### 3.16.3 Member Function Documentation



### 3.16.3.1 getTitulo()

```
string Titulo::getTitulo ( ) const [inline]
```

Funcao getTitulo, que instancia o objeto com uma string

#### Parameters

<i>string</i>	return string titulo
---------------	----------------------

### 3.16.3.2 setTitulo()

```
void Titulo::setTitulo (
    string titulo )
```

Funcao setTitulo, que instancia o objeto com uma string

#### Parameters

<i>string</i>	return void
---------------	-------------

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

- C:/Users/felip/Downloads/T1-TP1-200060295-202021767-202021800/dominios.h
- C:/Users/felip/Downloads/T1-TP1-200060295-202021767-202021800/dominios.cpp

## 3.17 TUAvaliacao Class Reference

### Public Member Functions

- int **run** ()

### Static Public Attributes

- static const int **SUCESSO** = 0
- static const int **FALHA** = -1

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

- C:/Users/felip/Downloads/T1-TP1-200060295-202021767-202021800/testes.h
- C:/Users/felip/Downloads/T1-TP1-200060295-202021767-202021800/testes.cpp

## 3.18 TUCidade Class Reference

### Public Member Functions

- int **run** ()

### Public Attributes

- int **estado**

### Static Public Attributes

- static const int **SUCESSO** = 0
- static const int **FALHA** = -1

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

- C:/Users/felip/Downloads/T1-TP1-200060295-202021767-202021800/testes.h
- C:/Users/felip/Downloads/T1-TP1-200060295-202021767-202021800/testes.cpp

## 3.19 TUCodigo Class Reference

### Public Member Functions

- `int run ()`

### Static Public Attributes

- static const int **SUCESSO** = 0
- static const int **FALHA** = -1

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

- C:/Users/felip/Downloads/T1-TP1-200060295-202021767-202021800/testes.h
- C:/Users/felip/Downloads/T1-TP1-200060295-202021767-202021800/testes.cpp

## 3.20 TUData Class Reference

### Public Member Functions

- `int run ()`

### Public Attributes

- `int estado`

### Static Public Attributes

- static const int **SUCESSO** = 0
- static const int **FALHA** = -1

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

- C:/Users/felip/Downloads/T1-TP1-200060295-202021767-202021800/testes.h
- C:/Users/felip/Downloads/T1-TP1-200060295-202021767-202021800/testes.cpp

## 3.21 TUDescricao Class Reference

### Public Member Functions

- `int run ()`

### Public Attributes

- `int estado`

### Static Public Attributes

- static const int **SUCESSO** = 0
- static const int **FALHA** = -1

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

- C:/Users/felip/Downloads/T1-TP1-200060295-202021767-202021800/testes.h
- C:/Users/felip/Downloads/T1-TP1-200060295-202021767-202021800/testes.cpp

## 3.22 TUDuracao Class Reference

### Public Member Functions

- `int run ()`

### Public Attributes

- `int estado`

### Static Public Attributes

- `static const int SUCESSO = 0`
- `static const int FALHA = -1`

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

- `C:/Users/felip/Downloads/T1-TP1-200060295-202021767-202021800/testes.h`
- `C:/Users/felip/Downloads/T1-TP1-200060295-202021767-202021800/testes.cpp`

## 3.23 TUEmail Class Reference

### Public Member Functions

- `int run ()`

### Public Attributes

- `int estado`

### Static Public Attributes

- `static const int SUCESSO = 0`
- `static const int FALHA = -1`

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

- `C:/Users/felip/Downloads/T1-TP1-200060295-202021767-202021800/testes.h`
- `C:/Users/felip/Downloads/T1-TP1-200060295-202021767-202021800/testes.cpp`

## 3.24 TUEndereco Class Reference

### Public Member Functions

- `int run ()`

### Public Attributes

- `int estado`

### Static Public Attributes

- `static const int SUCESSO = 0`
- `static const int FALHA = -1`

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

- `C:/Users/felip/Downloads/T1-TP1-200060295-202021767-202021800/testes.h`
- `C:/Users/felip/Downloads/T1-TP1-200060295-202021767-202021800/testes.cpp`

## 3.25 TUExcursao Class Reference

### Public Member Functions

- `int run ()`

### Static Public Attributes

- `static const int SUCESSO = 0`
- `static const int FALHA = -1`

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

- `C:/Users/felip/Downloads/T1-TP1-200060295-202021767-202021800/testes.h`
- `C:/Users/felip/Downloads/T1-TP1-200060295-202021767-202021800/testes.cpp`

## 3.26 TUHorario Class Reference

### Public Member Functions

- `int run ()`

### Static Public Attributes

- `static const int SUCESSO = 0`
- `static const int FALHA = -1`

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

- `C:/Users/felip/Downloads/T1-TP1-200060295-202021767-202021800/testes.h`
- `C:/Users/felip/Downloads/T1-TP1-200060295-202021767-202021800/testes.cpp`

## 3.27 TUIdioma Class Reference

### Public Member Functions

- `int run ()`

### Public Attributes

- `int estado`

### Static Public Attributes

- `static const int SUCESSO = 0`
- `static const int FALHA = -1`

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

- `C:/Users/felip/Downloads/T1-TP1-200060295-202021767-202021800/testes.h`
- `C:/Users/felip/Downloads/T1-TP1-200060295-202021767-202021800/testes.cpp`

## 3.28 TUNome Class Reference

### Public Member Functions

- `int run ()`

### Public Attributes

- int **estado**

### Static Public Attributes

- static const int **SUCESSO** = 0
- static const int **FALHA** = -1

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

- C:/Users/felip/Downloads/T1-TP1-200060295-202021767-202021800/testes.h
- C:/Users/felip/Downloads/T1-TP1-200060295-202021767-202021800/testes.cpp

## 3.29 TUNota Class Reference

### Public Member Functions

- int **run** ()

### Static Public Attributes

- static const int **SUCESSO** = 0
- static const int **FALHA** = -1

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

- C:/Users/felip/Downloads/T1-TP1-200060295-202021767-202021800/testes.h
- C:/Users/felip/Downloads/T1-TP1-200060295-202021767-202021800/testes.cpp

## 3.30 TUSenha Class Reference

### Public Member Functions

- int **run** ()

### Public Attributes

- int **estado**

### Static Public Attributes

- static const int **SUCESSO** = 0
- static const int **FALHA** = -1

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

- C:/Users/felip/Downloads/T1-TP1-200060295-202021767-202021800/testes.h
- C:/Users/felip/Downloads/T1-TP1-200060295-202021767-202021800/testes.cpp

## 3.31 TUSessao Class Reference

### Public Member Functions

- int **run** ()

### Static Public Attributes

- static const int **SUCESSO** = 0
- static const int **FALHA** = -1

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

- C:/Users/felip/Downloads/T1-TP1-200060295-202021767-202021800/testes.h
- C:/Users/felip/Downloads/T1-TP1-200060295-202021767-202021800/testes.cpp

## 3.32 TUTitulo Class Reference

### Public Member Functions

- int **run** ()

### Static Public Attributes

- static const int **SUCESSO** = 0
- static const int **FALHA** = -1

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

- C:/Users/felip/Downloads/T1-TP1-200060295-202021767-202021800/testes.h
- C:/Users/felip/Downloads/T1-TP1-200060295-202021767-202021800/testes.cpp

## 3.33 TUUsuario Class Reference

### Public Member Functions

- int **run** ()

### Static Public Attributes

- static const int **SUCESSO** = 0
- static const int **FALHA** = -1

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

- C:/Users/felip/Downloads/T1-TP1-200060295-202021767-202021800/testes.h
- C:/Users/felip/Downloads/T1-TP1-200060295-202021767-202021800/testes.cpp

## 3.34 Usuario Class Reference

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

### Public Member Functions

- void **setNome** (const **Nome** &)
- void **setEmail** (const **Email** &)
- void **setSenha** (const **Senha** &)
- **Nome** **getNome** () const
- **Email** **getEmail** () const
- **Senha** **getSenha** () const

### 3.34.1 Detailed Description

3.34.1.1 Declaracao da entidade Usuario que permite instanciar um nome, email e senha.

---

### 3.34.2 Member Function Documentation

#### 3.34.2.1 getEmail()

```
Email Usuario::getEmail ( ) const [inline]
```

Metodo que recupera o email return string email

#### 3.34.2.2 getNome()

```
Nome Usuario::getNome ( ) const [inline]
```

Metodo que recupera o nome return string nome

#### 3.34.2.3 getSenha()

```
Senha Usuario::getSenha ( ) const [inline]
```

Metodo que recupera a senha return string senha

#### 3.34.2.4 setEmail()

```
void Usuario::setEmail (
    const Email & email ) [inline]
```

Metodo que seta o email

##### Parameters

<i>string</i>	email return void
---------------	-------------------

#### 3.34.2.5 setName()

```
void Usuario::setName (
    const Nome & nome ) [inline]
```

Metodo que seta o nome

##### Parameters

<i>string</i>	nome return void
---------------	------------------

#### 3.34.2.6 setSenha()

```
void Usuario::setSenha (
    const Senha & senha ) [inline]
```

Metodo que seta o nome

##### Parameters

<i>string</i>	nome return void
---------------	------------------

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

- C:/Users/felip/Downloads/T1-TP1-200060295-202021767-202021800/entidades.h





## Chapter 4

# File Documentation

### 4.1 dominios.h

```
1 #ifndef DOMINIOS_H_INCLUDED
2 #define DOMINIOS_H_INCLUDED
3
4 #include <stdexcept>
5 #include <iostream>
6 #include <string>
7
8 #define ll long long;
9
10 using namespace std;
11
12 // Matricula: 200060295.
13 class Email {
14     private:
15     // @param string email
16     string email;
17     const static int TAMANHO_MAX_LOCAL = 64; // constante
18     const static int TAMANHO_MAX_DOMINIO = 253; // constante
19
20     // Funcao validar que valida uma string que sera usada como email
21     // @param string
22     // @throw invalid_argument
23     // @return void
24     void validar(string) throw (invalid_argument);
25
26     public:
27     Email(); // default
28
29     Email(string);
30
31     void setEmail(string);
32
33     string getEmail() const {
34         return email;
35     }
36 };
37
38 // Matricula: 200060295.
39 class Nota {
40     private:
41     // @param int nota
42     int nota;
43
44     // Funcao validar que valida um inteiro
45     // @throw invalid_argument
46     // @return void
47     void validar(int) throw (invalid_argument);
48
49     public:
50     Nota(); // default
51
52     Nota(int);
53
54     void setNota(int);
55
56     int getNota() const {
57         return nota;
58     }
59 };
60
61 // Matricula: 200060295.
62 class Nome {
```

```
88     private:
89     // @param string nome
90     string nome;
91     const static unsigned int TAMANHO_MIN = 5; //constante
92     const static unsigned int TAMANHO_MAX = 20; //constante
93
94     // Funcao validar que valida uma string
95     // @throw invalid_argument
96     // @return void
97     void validar(string) throw (invalid_argument);
98
99     public:
100     Nome(); // default
101
102     Nome(string);
103
104     void setNome(string);
105
106     string getNome() const {
107         return nome;
108     }
109 };
110 // Matricula: 202021767.
111 class Senha {
112     private:
113     // @param string senha
114     string senha;
115     const static unsigned int LIMITE = 6; //constante
116
117     // Funcao validar que valida uma string
118     // @throw invalid_argument
119     // @return void
120     void validar(string) throw (invalid_argument);
121
122     public:
123     Senha(); // default
124
125     Senha(string);
126
127     void setSenha(string);
128
129     string getSenha() const {
130         return senha;
131     }
132 };
133 // Matricula: 200060295.
134 class Data {
135     private:
136     // @param string data
137     string data;
138     const static unsigned int TAMANHO_MAX = 11; //constante
139
140     // Funcao validar que valida uma string
141     // @throw invalid_argument
142     // @return void
143     void validar(string) throw (invalid_argument);
144
145     public:
146     Data(); // default
147
148     Data(string);
149
150     void setData(string);
151
152     string getData() const {
153         return data;
154     }
155 };
156 // Matricula: 200060295.
157 class Horario {
158     private:
159     // @param string horario
160     string horario;
161     // Funcao validar que valida uma string
162     // @throw invalid_argument
163     // @return void
164     void validar(string) throw (invalid_argument);
165     const static unsigned int TAMANHO_MAX = 5;
166
167     public:
168     Horario(); // default
169
170     Horario(string);
171
172     void setHorario(string);
173
174     string getHorario() const {
```

```
217         return horario;
218     }
219 };
220
221 // Matricula: 200060295.
222 class Idioma {
223 private:
224     // @param string idioma
225     string idioma;
226     // Funcao validar que valida uma string
227     // @throw invalid_argument
228     // @return void
229     void validar(string) throw (invalid_argument);
230
231 public:
232     Idioma(); // default
233     Idioma(string);
234     void setIdioma(string);
235     string getIdioma() const {
236         return idioma;
237     }
238 };
239
240 // Matricula: 200060295.
241 class Titulo {
242 private:
243     // @param string titulo
244     string titulo;
245     const static int TAMANHO_MIN = 5;
246     const static int TAMANHO_MAX = 20;
247     // Funcao validar que valida uma string
248     // @throw invalid_argument
249     // @return void
250     void validar(string) throw (invalid_argument);
251
252 public:
253     Titulo(); // default
254     Titulo(string);
255     void setTitulo(string);
256     string getTitulo() const {
257         return titulo;
258     }
259 };
260
261 // Matricula: 200060295.
262 class Cidade {
263 private:
264     // @param string cidade
265     string cidade;
266     // Funcao validar que valida uma string
267     // @throw invalid_argument
268     // @return void
269     void validar(string) throw (invalid_argument);
270
271 public:
272     Cidade(); // default
273     Cidade(string);
274     void setCidade(string);
275     string getCidade() const {
276         return cidade;
277     }
278 };
279
280 // Matricula: 200060295.
281 class Duracao {
282 private:
283     // @param string duracao
284     string duracao;
285     // Funcao validar que valida uma string
286     // @throw invalid_argument
287     // @return void
288     void validar(string) throw (invalid_argument);
289
290 public:
291     Duracao(); // default
292     Duracao(string);
293     void setDuracao(string);
294     string getDuracao() const {
295         return duracao;
296     }
297 };
298
299 // Matricula: 200060295.
300 class Descricao {
301 private:
302     // @param string descricao
303     string descricao;
304     const static int TAMANHO_MIN = 0;
```

```

355     const static int TAMANHO_MAX = 30;
356 // Funcao validar que valida uma string
357 // @throw invalid_argument
358 // @return void
359     void validar(string) throw (invalid_argument);
360
361 public:
362     Descricao(); // default
363     Descricao(string);
364     void setDescricao(string);
365     string getDescricao() const {
366         return descricao;
367     }
368 };
369
370 // Matricula: 200060295.
371 class Codigo {
372 private:
373     // @param string codigo
374     string codigo;
375     const static unsigned int TAMANHO_MAX = 7;
376 // Funcao validar que valida uma string
377 // @throw invalid_argument
378 // @return void
379     void validar(string) throw (invalid_argument);
380
381 public:
382     Codigo(); // default
383     Codigo(string);
384     void setCodigo(string);
385     string getCodigo() const {
386         return codigo;
387     }
388 };
389
390 // Matricula: 200060295.
391 class Endereco {
392 private:
393     // @param string endereco
394     string endereco;
395     const static int TAMANHO_MIN = 0;
396     const static int TAMANHO_MAX = 20;
397 // Funcao validar que valida uma string
398 // @throw invalid_argument
399 // @return void
400     void validar(string) throw (invalid_argument);
401
402 public:
403     Endereco(); // default
404     Endereco(string);
405     void setEndereco(string);
406     string getEndereco() const {
407         return endereco;
408     }
409 };
410
411 #endif // DOMINIOS_H_INCLUDED

```

## 4.2 entidades.h

```

1 #ifndef ENTIDADES_H_INCLUDED
2 #define ENTIDADES_H_INCLUDED
3
4 #include <iostream>
5 #include <iomanip>
6 #include <math.h>
7 #include <string>
8 #include <bits/stdc++.h>
9
10 #include "dominios.h"
11
12 using namespace std;
13
14 // Matricula: 200060295
15
16 class Usuario {
17 private:
18     Nome nome;
19     Email email;
20     Senha senha;
21
22 public:
23     void setNome(const Nome&);
24     void setEmail(const Email&);

```

```

33         void setSenha(const Senha&);
34
35         Nome getName() const;
36         Email getEmail() const;
37         Senha getSenha() const;
38     };
39
40     // -----
41     // Implementacao Metodos inline set da entidade Usuario.
42     // -----
43
44 inline void Usuario::setName(const Nome& nome){
45     this->nome = nome;
46 }
47 inline void Usuario::setEmail(const Email& email){
48     this->email = email;
49 }
50 inline void Usuario::setSenha(const Senha& senha){
51     this->senha = senha;
52 }
53
54 // -----
55 // Implementacao Metodos inline get da entidade Usuario.
56 // -----
57
58 inline Nome Usuario::getName() const {
59     return nome;
60 }
61 inline Email Usuario::getEmail() const {
62     return email;
63 }
64 inline Senha Usuario::getSenha() const {
65     return senha;
66 }
67
68 // Matricula: 200060295
69
70 class Avaliacao {
71     private:
72
73         Codigo codigo;
74         Nota nota;
75         Descricao descricao;
76     public:
77         void setCodigo(const Codigo&);
78         void setNota(const Nota&);
79         void setDescricao(const Descricao&);
80
81         Codigo getCodigo() const;
82         Nota getNota() const;
83         Descricao getDescricao() const;
84 };
85
86 inline void Avaliacao::setCodigo(const Codigo& codigo){
87     this->codigo = codigo;
88 }
89 inline void Avaliacao::setDescricao(const Descricao& descricao){
90     this->descricao = descricao;
91 }
92 inline void Avaliacao::setNota(const Nota& nota){
93     this->nota = nota;
94 }
95
96 // -----
97 // Implementacao Metodos inline get da entidade Avaliacao.
98 // -----
99
100 inline Codigo Avaliacao::getCodigo() const {
101     return codigo;
102 }
103 inline Nota Avaliacao::getNota() const {
104     return nota;
105 }
106 inline Descricao Avaliacao::getDescricao() const {
107     return descricao;
108 }
109
110 // Matricula: 202021767
111
112 class Excursao {
113     private:
114
115         Codigo codigo;
116         Titulo titulo;
117         Nota nota;
118         Cidade cidade;
119         Duracao duracao;
120         Descricao descricao;

```

```

169     Endereco endereco;
170 public:
171     void setCodigo(const Codigo&);
172     void setTitulo(const Titulo&);
173     void setNota(const Nota&);
174     void setCidade(const Cidade&);
175     void setDuracao(const Duracao&);
176     void setDescricao(const Descricao&);
177     void setEndereco(const Endereco&);
178
179     Codigo getCodigo() const;
180     Titulo getTitulo() const;
181     Nota getNota() const;
182     Cidade getCidade() const;
183     Duracao getDuracao() const;
184     Descricao getDescricao() const;
185     Endereco getEndereco() const;
186 };
187 // -----
188 // Implementacao Metodos inline set da entidade Excursao.
189 // -----
190
191 inline void Excursao::setCodigo(const Codigo& codigo){
192     this->codigo = codigo;
193 }
194
195 inline void Excursao::setTitulo(const Titulo& titulo){
196     this->titulo = titulo;
197 }
198
199 inline void Excursao::setNota(const Nota& nota){
200     this->nota = nota;
201 }
202
203 inline void Excursao::setCidade(const Cidade& cidade){
204     this->cidade = cidade;
205 }
206
207 inline void Excursao::setDuracao(const Duracao& duracao){
208     this->duracao = duracao;
209 }
210
211 inline void Excursao::setDescricao(const Descricao& descricao){
212     this->descricao = descricao;
213 }
214
215 inline void Excursao::setEndereco(const Endereco& endereco){
216     this->endereco = endereco;
217 }
218 // -----
219 // Implementacao Metodos inline get da entidade Excursao.
220 // -----
221
222 inline Codigo Excursao::getCodigo() const {
223     return codigo;
224 }
225
226 inline Titulo Excursao::getTitulo() const {
227     return titulo;
228 }
229
230 inline Nota Excursao::getNota() const {
231     return nota;
232 }
233
234 inline Cidade Excursao::getCidade() const {
235     return cidade;
236 }
237
238 inline Duracao Excursao::getDuracao() const {
239     return duracao;
240 }
241
242 inline Descricao Excursao::getDescricao() const {
243     return descricao;
244 }
245
246 inline Endereco Excursao::getEndereco() const {
247     return endereco;
248 }
249 // Matricula: 202021767
250 class Sessao {
251 private:
252
253     Codigo codigo;
254     Data data;
255     Horario horario;
256     Idioma idioma;
257 public:
258     void setCodigo(const Codigo&);
259     void setData(const Data&);
260     void setHorario(const Horario&);
261     void setIdioma(const Idioma&);
262
263     Codigo getCodigo() const;
264     Data getData() const;
265     Horario getHorario() const;
266     Idioma getIdioma() const;

```

```

298 };
299 // -----
300 // Implementacao Metodos inline set da entidade Sessao.
301 // -----
302
306 inline void Sessao::setCodigo(const Codigo& codigo){
307     this->codigo = codigo;
308 }
312 inline void Sessao::setData(const Data& data){
313     this->data = data;
314 }
318 inline void Sessao::setHorario(const Horario& horario){
319     this->horario = horario;
320 }
321 }
325 inline void Sessao::setIdioma(const Idioma& idioma){
326     this->idioma = idioma;
327 }
328 // -----
329 // Implementação Metodos inline get da entidade Sessao.
330 // -----
331
334 inline Codigo Sessao::getCodigo() const {
335     return codigo;
336 }
339 inline Data Sessao::getData() const {
340     return data;
341 }
344 inline Horario Sessao::getHorario() const {
345     return horario;
346 }
349 inline Idioma Sessao::getIdioma() const {
350     return idioma;
351 }
352
353 #endif // ENTIDADES_H_INCLUDED
354

```

## 4.3 testes.h

```

1 #ifndef TESTES_H_INCLUDED
2 #define TESTES_H_INCLUDED
3
4 #include <string>
5 #include <bits/stdc++.h>
6 #include "dominios.h"
7 #include "entidades.h"
8
9 using namespace std;
10
11 // -----
12 // Implementacao classes de teste de unidade de dominios (TUCodigo, TUTitulo, TUHorario, TUNota).
13 // Matricula: 200060295
14 // -----
15
16 class TUCodigo {
17     private:
18         string VALOR_VALIDO = "0000026";
19         string VALOR_INVALIDO = "0000020";
20         Codigo *codigo;
21         int estado;
22         void setUp();
23         void tearDown();
24         void testarCenarioSucesso();
25         void testarCenarioFalha();
26
27     public:
28         const static int SUCESSO = 0;
29         const static int FALHA = -1;
30         int run();
31 };
32
33 class TUTitulo {
34     private:
35         string VALOR_VALIDO = "Ex. de Titulo valido";
36         string VALOR_INVALIDO = "Ex.. de Titulo invalido";
37         Titulo *titulo;
38         int estado;
39         void setUp();
40         void tearDown();
41         void testarCenarioSucesso();
42         void testarCenarioFalha();
43
44     public:
45         const static int SUCESSO = 0;

```

```

46     const static int FALHA = -1;
47     int run();
48 };
49
50 class TUHorario {
51 private:
52     string VALOR_VALIDO = "04:58";
53     string VALOR_INVALIDO = "04:65";
54     Horario *horario;
55     int estado;
56     void setUp();
57     void tearDown();
58     void testarCenarioSucesso();
59     void testarCenarioFalha();
60
61 public:
62     const static int SUCESSO = 0;
63     const static int FALHA = -1;
64     int run();
65 };
66
67 class TUNota {
68 private:
69     int VALOR_VALIDO = 5;
70     int VALOR_INVALIDO = 9;
71     Nota *nota;
72     int estado;
73     void setUp();
74     void tearDown();
75     void testarCenarioSucesso();
76     void testarCenarioFalha();
77
78 public:
79     const static int SUCESSO = 0;
80     const static int FALHA = -1;
81     int run();
82 };
83
84 // -----
85 // Implementacao classes de teste de unidade de dominios (TUEmail, TUData, TUDescricao, TUEndereo,
86 // TUIdioma).
87 // Matricula: 202021800
88 // -----
89 class TUEmail {
90 private:
91     const string EMAIL_VALIDO = "aquilamacedo@riseup.net";
92     const string EMAIL_INVALIDO = "Aq$u114{m#ce$+=do.@rlsE-upnet";
93     Email *email;
94     void setUp();
95     void tearDown();
96     void testarCenarioSucesso();
97     void testarCenarioFalha();
98
99 public:
100     const static int SUCESSO = 0;
101     const static int FALHA = -1;
102     int estado;
103     int run();
104 };
105
106 class TUData {
107 private:
108     const string DATA_VALIDA = "29-Fev-2020";
109     const string DATA_INVALIDA = "29/Fev/2020";
110     Data *data;
111     void setUp();
112     void tearDown();
113     void testarCenarioSucesso();
114     void testarCenarioFalha();
115
116 public:
117     const static int SUCESSO = 0;
118     const static int FALHA = -1;
119     int estado;
120     int run();
121 };
122
123 class TUDescricao {
124 private:
125     const string DESCRICAO_VALIDA = "aaaaaaaaaaaaaaaaaaaaaaaaaaaa";
126     const string DESCRICAO_INVALIDA = "zzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzz";
127     Descricao *descricao;
128     void setUp();
129     void tearDown();
130     void testarCenarioSucesso();
131     void testarCenarioFalha();

```



```
132
133 public:
134     const static int SUCESSO = 0;
135     const static int FALHA = -1;
136     int estado;
137     int run();
138 };
139
140 class TUEndereco {
141 private:
142     const string ENDERECO_VALIDO = "Toronto, Canada";
143     const string ENDERECO_INVALIDO = "Cidade do Mexico, Mexico";
144     Endereco *endereco;
145     void setUp();
146     void tearDown();
147     void testarCenarioSucesso();
148     void testarCenarioFalha();
149
150 public:
151     const static int SUCESSO = 0;
152     const static int FALHA = -1;
153     int estado;
154     int run();
155 };
156
157 class TUIdioma {
158 private:
159     const string IDIOMA_VALIDO = "Portugues";
160     const string IDIOMA_INVALIDO = "Japones";
161     Idioma *idioma;
162     void setUp();
163     void tearDown();
164     void testarCenarioSucesso();
165     void testarCenarioFalha();
166
167 public:
168     const static int SUCESSO = 0;
169     const static int FALHA = -1;
170     int estado;
171     int run();
172 };
173
174 // -----
175 // Implementacao classes de teste de unidade de dominios TUNome.
176 // Matricula: 202021767
177 // -----
178
179 class TUNome {
180 private:
181     string NOME_VALIDO = "Felipe";
182     string NOME_INVALIDO = ".fel";
183     Nome *nome;
184     void setUp();
185     void tearDown();
186     void testarCenarioSucesso();
187     void testarCenarioFalha();
188 public:
189     const static int SUCESSO = 0;
190     const static int FALHA = -1;
191     int estado;
192     int run();
193 };
194
195 // -----
196 // Implementacao classes de teste de unidade de dominios TUSenha.
197 // Matricula: 202021767
198 // -----
199
200 class TUSenha {
201 private:
202     string SENHA_VALIDA = "F3l1pe";
203     string SENHA_INVALIDA = "FELPPP";
204     Senha *senha;
205     void setUp();
206     void tearDown();
207     void testarCenarioSucesso();
208     void testarCenarioFalha();
209 public:
210     const static int SUCESSO = 0;
211     const static int FALHA = -1;
212     int estado;
213     int run();
214 };
215
216 // -----
217 // Implementacao classes de teste de unidade de dominios TUCidade.
218 // Matricula: 202021767
```

```

219 // -----
220
221 class TUCidade {
222     private:
223         string CIDADE_VALIDA = "Bangkok";
224         string CIDADE_INVALIDA = "Asa Norte";
225         Cidade *cidade;
226         void setUp();
227         void tearDown();
228         void testarCenarioSucesso();
229         void testarCenarioFalha();
230     public:
231         const static int SUCESSO = 0;
232         const static int FALHA = -1;
233         int estado;
234         int run();
235 };
236
237 // -----
238 // Implementacao classes de teste de unidade de dominios TUDuracao.
239 // Matricula: 202021767
240 // -----
241
242 class TUDuracao {
243     private:
244         string DURACAO_VALIDA = "180";
245         string DURACAO_INVALIDA = "200";
246         Duracao *duracao;
247         void setUp();
248         void tearDown();
249         void testarCenarioSucesso();
250         void testarCenarioFalha();
251     public:
252         const static int SUCESSO = 0;
253         const static int FALHA = -1;
254         int estado;
255         int run();
256 };
257
258 // -----
259 // Implementacao classes de teste de unidade de entidade TUUsuario.
260 // Matricula: 200060295
261 // -----
262
263 class TUUsuario {
264     private:
265         string VALOR_VALIDO_NOME = "Felipe";
266         string VALOR_INVALIDO_NOME = ".fel";
267         string VALOR_VALIDO_EMAIL = "teste@unb.com.br";
268         string VALOR_INVALIDO_EMAIL = "@teste@.com.br";
269         string VALOR_VALIDO_SENHA = "F3lipse";
270         string VALOR_INVALIDO_SENHA = "FELPPP";
271         Usuario *usuario;
272         int estado;
273         void setUp();
274         void tearDown();
275         void testar();
276     public:
277         const static int SUCESSO = 0;
278         const static int FALHA = -1;
279         int run();
280 };
281
282 // -----
283 // Implementacao classes de teste de unidade de entidade TUAvaliacao.
284 // Matricula: 200060295
285 // -----
286
287 class TUAvaliacao {
288     private:
289         string VALOR_VALIDO_CODIGO = "0000026";
290         string VALOR_INVALIDO_CODIGO = "0000020";
291         const static int VALOR_VALIDO_NOTA = 2;
292         const static int VALOR_INVALIDO_NOTA = 1000000;
293         string VALOR_VALIDO_DESCRICAO = "Isso eh uma descricao";
294         string VALOR_INVALIDO_DESCRICAO = " @!#Nao eh ..";
295         Avaliacao *avaliacao;
296         int estado;
297         void setUp();
298         void tearDown();
299         void testar();
300     public:
301         const static int SUCESSO = 0;
302         const static int FALHA = -1;
303         int run();
304 };

```

```
306 };
307
308 // -----
309 // Implementacao classes de teste de unidade de entidade TUExcursao.
310 // Matricula: 202021800
311 // -----
312
313 class TUExcursao {
314     private:
315         string VALOR_VALIDO_CODIGO = "0000026";
316         string VALOR_INVALIDO_CODIGO = "0000020";
317         string VALOR_VALIDO_TITULO = "Git Workflow";
318         string VALOR_INVALIDO_TITULO = "InV!";
319         const static int VALOR_VALIDO_NOTA = 3;
320         const static int VALOR_INVALIDO_NOTA = 1337;
321         string VALOR_VALIDO_CIDADE = "Macau";
322         string VALOR_INVALIDO_CIDADE = "Toronto";
323         string VALOR_VALIDO_DURACAO = "120";
324         string VALOR_INVALIDO_DURACAO = "1999";
325         string VALOR_VALIDO_DESCRICAO = "When see a good move";
326         string VALOR_INVALIDO_DESCRICAO = "When you see a good move look for a better one";
327         string VALOR_VALIDO_ENDERECO = "Toronto, Canada";
328         string VALOR_INVALIDO_ENDERECO = "Toronto Canada..";
329         Excursao *excursao;
330         int estado;
331         void setUp();
332         void tearDown();
333         void testar();
334
335     public:
336         const static int SUCESSO = 0;
337         const static int FALHA = -1;
338         int run();
339 };
340
341 // -----
342 // Implementacao classes de teste de unidade de entidade TUSessao.
343 // Matricula: 202021800
344 // -----
345
346 class TUSessao {
347     private:
348         string VALOR_VALIDO_CODIGO = "1929827";
349         string VALOR_INVALIDO_CODIGO = "0000020";
350         string VALOR_VALIDO_DATA = "29-Fev-2020";
351         string VALOR_INVALIDO_DATA = "29/Fev/2222";
352         string VALOR_VALIDO_HORARIO = "23:00";
353         string VALOR_INVALIDO_HORARIO = "23h00";
354         string VALOR_VALIDO_IDIOMA = "Portugues";
355         string VALOR_INVALIDO_IDIOMA = "Alemao";
356         Sessao *sessao;
357         int estado;
358         void setUp();
359         void tearDown();
360         void testar();
361
362     public:
363         const static int SUCESSO = 0;
364         const static int FALHA = -1;
365         int run();
366 };
367
368 #endif
```



# Index

Avaliacao, [5](#)  
    getCodigo, [5](#)  
    getDescricao, [5](#)  
    getNota, [5](#)  
    setCodigo, [5](#)  
    setDescricao, [7](#)  
    setNota, [7](#)

C:/Users/felip/Downloads/T1-TP1-200060295-202021767-202021800/dominios.h, [33](#)

C:/Users/felip/Downloads/T1-TP1-200060295-202021767-202021800/entidades.h, [36](#)

C:/Users/felip/Downloads/T1-TP1-200060295-202021767-202021800/testes.h, [39](#)

Cidade, [7](#)  
    Cidade, [7](#)  
    getCidade, [8](#)  
    setCidade, [8](#)

Codigo, [8](#)  
    Codigo, [8](#)  
    getCodigo, [9](#)  
    setCodigo, [9](#)

Data, [9](#)  
    Data, [10](#)  
    getData, [10](#)  
    setData, [10](#)

Descricao, [10](#)  
    Descricao, [11](#)  
    getDescricao, [11](#)  
    setDescricao, [11](#)

Duracao, [11](#)  
    Duracao, [12](#)  
    getDuracao, [12](#)  
    setDuracao, [12](#)

Email, [12](#)  
    Email, [13](#)  
    getEmail, [13](#)  
    setEmail, [13](#)

Endereco, [13](#)  
    Endereco, [14](#)  
    getEndereco, [14](#)  
    setEndereco, [14](#)

Excursao, [15](#)  
    getCidade, [15](#)  
    getCodigo, [15](#)  
    getDescricao, [15](#)  
    getDuracao, [15](#)  
    getEndereco, [15](#)  
    getNota, [15](#)  
    getTitulo, [15](#)  
    setCidade, [16](#)  
    setCodigo, [16](#)  
    setDescricao, [16](#)  
    setDuracao, [16](#)  
    setEndereco, [16](#)  
    setNota, [17](#)  
    setTitulo, [17](#)

Excursao, [15](#)  
    getCodigo  
        Avaliacao, [5](#)  
        Codigo, [9](#)  
        Excursao, [15](#)  
        Sessao, [23](#)  
    getData  
        Data, [10](#)  
        Sessao, [23](#)  
    getDescricao  
        Avaliacao, [5](#)  
        Descricao, [11](#)  
        Excursao, [15](#)  
    getDuracao  
        Duracao, [12](#)  
        Excursao, [15](#)  
    getEmail  
        Email, [13](#)  
        Usuario, [31](#)  
    getEndereco  
        Endereco, [14](#)  
        Excursao, [15](#)  
    getHorario  
        Horario, [18](#)  
        Sessao, [23](#)  
    getIdioma  
        Idioma, [19](#)  
        Sessao, [23](#)  
    getNome  
        Nome, [20](#)  
        Usuario, [31](#)  
    getNota  
        Avaliacao, [5](#)  
        Excursao, [15](#)  
        Nota, [21](#)  
    getSenha  
        Senha, [22](#)  
        Usuario, [31](#)

getTitulo  
Excursao, 15  
Titulo, 24

Horario, 17  
getHorario, 18  
Horario, 17  
setHorario, 18

Idioma, 18  
getIdioma, 19  
Idioma, 18  
setIdioma, 19

Nome, 19  
getNome, 20  
Nome, 20  
setNome, 20

Nota, 20  
getNota, 21  
Nota, 21  
setNota, 21

Senha, 21  
getSenha, 22  
Senha, 22  
setSenha, 22

Sessao, 22  
getCodigo, 23  
getData, 23  
getHorario, 23  
getIdioma, 23  
setCodigo, 23  
setData, 23  
setHorario, 23  
setIdioma, 24

setCidade  
Cidade, 8  
Excursao, 16

setCodigo  
Avaliacao, 5  
Codigo, 9  
Excursao, 16  
Sessao, 23

setData  
Data, 10  
Sessao, 23

setDescricao  
Avaliacao, 7  
Descricao, 11  
Excursao, 16

setDuracao  
Duracao, 12  
Excursao, 16

setEmail  
Email, 13  
Usuario, 31

setEndereco  
Endereco, 14  
Excursao, 16

setHorario  
Horario, 18  
Sessao, 23

setIdioma  
Idioma, 19  
Sessao, 24

setNome  
Nome, 20  
Usuario, 31

setNota  
Avaliacao, 7  
Excursao, 17  
Nota, 21

setSenha  
Senha, 22  
Usuario, 31

setTitulo  
Excursao, 17  
Titulo, 25

Titulo, 24  
getTitulo, 24  
setTitulo, 25  
Titulo, 24

TUAvaliacao, 25

TUCidade, 25

TUCodigo, 26

TUData, 26

TUDescricao, 26

TUDuracao, 27

TUEmail, 27

TUEndereco, 27

TUExcursao, 28

TUHorario, 28

TUIdioma, 28

TUNome, 28

TUNota, 29

TUSenha, 29

TUSessao, 29

TUTitulo, 30

TUUsuario, 30

Usuario, 30  
getEmail, 31  
getNome, 31  
getSenha, 31  
setEmail, 31  
setNome, 31  
setSenha, 31