

TP PDS2
Gerenciador de Tarefas

Ricardo Barbosa, Igor Henrique e Victor Seabra
Versão 1.0

06/13/2019

2019-1-grupo17

Grupo: Ricardo

Tema: Gerenciador de um Instituto de Ensino: Criar um gerenciador para uma instituição de Ensino de pequeno/médio porte, em que a instituição possa cadastrar, professores, alunos, discente, docente e disciplinas.

Programa implementado em C++. Testado no linux, com todas funcionalidades aprovadas.

Maiores detalhes sobre a implementação, como estruturas de dados utilizadas, detalhes de funcionalidade e compilação estão descritas na documentação.

Arquivos separados em diretórios /src/ - arquivos .cpp /include/ - arquivos .hpp /documentation/ - arquivos de documentação e readme /build/ - arquivos.o

Team 17

Eu como instituição de educação quero gerir pessoas, disciplinas, departamentos.

- Cadastrar Professores, Alunos, Disciplinas, Departamentos.
- Excluir Professores, Alunos, Disciplinas, Departamentos.
- Alterar Professores, Alunos, Disciplinas, Departamentos.

Eu como instituição de educação quero um sistema para monitoramento interno.

- Pesquisar Professores (Disciplinas, **Departamento**)
- Pesquisar Alunos (Disciplinas)
- Pesquisar Disciplinas (Professores, Alunos, **Departamento**)
- Pesquisar Departamentos (Disciplinas)

Eu como instituição de educação quero visualizar todas as entidades.

- Listar Professores
- Listar Alunos
- Listar Disciplinas
- Listar Departamentos

Eu como instituição de educação quero gerir os alunos nas disciplinas.

- Cadastrar **Aluno** na **Disciplina**
- Excluir **Aluno** da **Disciplina**
- Trancar **Disciplina** para o **Aluno**

Eu como instituição de educação quero gerir os professores nas disciplinas.

- Cadastrar **Professor** na **Disciplina**
- Excluir **Professor** na **Disciplina**

Eu como instituição de educação quero gerir as disciplinas nos departamentos.

- Cadastrar **Disciplina** no **Departamento**

- Excluir **Disciplina** do **Departamento**

Eu como instituição de educação quero um sistema de notas para os alunos.

- Cadastrar Notas de Alunos
- Editar Notas de Alunos
- Aprovar/Reprovar Alunos

Índice da hierarquia

Hierarquia de classes

Esta lista de heranças está organizada, dentro do possível, por ordem alfabética:

Departamento.....	9
Disciplina.....	11
Pessoa.....	13
Discente.....	10
Aluno.....	8
Docente.....	12
Professor.....	15

Índice dos componentes

Lista de componentes

Lista de classes, estruturas, uniões e interfaces com uma breve descrição:

Aluno	8
Departamento	9
Discente	10
Disciplina	11
Docente	12
Pessoa	13
Professor	15

Índice dos ficheiros

Lista de ficheiros

Lista de todos os ficheiros com uma breve descrição:

20191-team-17-ricardo/include/aluno.hpp (Class Aluno)	16
20191-team-17-ricardo/include/departamento.hpp (Class Departamento)	17
20191-team-17-ricardo/include/discente.hpp (Class Discente)	18
20191-team-17-ricardo/include/disciplina.hpp (Class Disciplina)	19
20191-team-17-ricardo/include/docente.hpp (Class Docente)	20
20191-team-17-ricardo/include/pessoa.hpp (Class Pessoa)	21
20191-team-17-ricardo/include/professor.hpp (Class Professor)	22
20191-team-17-ricardo/src/aluno.cpp	24
20191-team-17-ricardo/src/departamento.cpp	25
20191-team-17-ricardo/src/disciplina.cpp	26
20191-team-17-ricardo/src/main.cpp	27
20191-team-17-ricardo/src/professor.cpp	31

Documentação da classe

Referência à classe Aluno

```
#include <aluno.hpp>
```

Diagrama de heranças da classe Aluno

Membros públicos

- **Aluno** (std::string nome, std::string cpf, std::string data_nasc, std::string endereco, double matricula)
 - double **get_matricula** ()
-

Documentação dos Construtores & Destrutor

Aluno::Aluno (std::string *nome*, std::string *cpf*, std::string *data_nasc*, std::string *endereco*, double *matricula*)

```
13                                     : Discente(nome, cpf, data_nasc, endereco){  
14     _matricula = matricula;  
15 }
```

Documentação dos métodos

double Aluno::get_matricula () [virtual]

Implementa **Discente** (*p. 11*).

```
17                                     {  
18     return _matricula;  
19 }
```

A documentação para esta classe foi gerada a partir dos seguintes ficheiros:

- 20191-team-17-ricardo/include/**aluno.hpp**
- 20191-team-17-ricardo/src/**aluno.cpp**

Referência à classe Departamento

```
#include <departamento.hpp>
```

Membros públicos

- **Departamento** (std::string nome, double num_funcionarios)
 - std::string **get_nome** ()
 - double **get_num_funcionarios** ()
-

Documentação dos Construtores & Destrutor

Departamento::Departamento (std::string *nome*, double *num_funcionarios*) [inline]

```
25                                     {
26         _nome = nome;
27         _num_funcionarios = num_funcionarios;
28     }
```

Documentação dos métodos

std::string Departamento::get_nome () [inline]

```
29     {
30         return _nome;
31     };
```

double Departamento::get_num_funcionarios () [inline]

```
32     {
33         return _num_funcionarios;
34     };
```

A documentação para esta classe foi gerada a partir do seguinte ficheiro:

- 20191-team-17-ricardo/include/**departamento.hpp**

Referência à classe Discente

```
#include <discente.hpp>
```

Diagrama de heranças da classe Discente

Membros públicos

- **Discente** (std::string nome, std::string cpf, std::string data_nasc, std::string endereco)
- virtual double **get_matricula** ()=0

Documentação dos Construtores & Destrutor

```
Discente::Discente (std::string nome, std::string cpf, std::string data_nasc,  
std::string endereco)[inline]
```

```
    27                                     : Pessoa(nome, cpf, data_nasc, endereco)  
    {};
```

Documentação dos métodos

```
virtual double Discente::get_matricula () [pure virtual]
```

Implementado em **Aluno** (p. 9).

A documentação para esta classe foi gerada a partir do seguinte ficheiro:

- 20191-team-17-ricardo/include/discente.hpp

Referência à classe Disciplina

```
#include <disciplina.hpp>
```

Membros públicos

- **Disciplina** (std::string nome, std::string semestre, double num_alunos)
 - std::string **get_nome** ()
 - std::string **get_semestre** ()
 - double **get_num_alunos** ()
-

Documentação dos Construtores & Destrutor

Disciplina::Disciplina (std::string *nome*, std::string *semestre*, double *num_alunos*)
[inline]

```
27                                     {  
28         _nome = nome;  
29         _semestre = semestre;  
30         _num_alunos = num_alunos;  
31     }
```

Documentação dos métodos

std::string Disciplina::get_nome () [inline]

```
32     {  
33         return _nome;  
34     };
```

double Disciplina::get_num_alunos () [inline]

```
38     {  
39         return _num_alunos;  
40     };
```

std::string Disciplina::get_semestre () [inline]

```
35     {  
36         return _semestre;  
37     };
```

A documentação para esta classe foi gerada a partir do seguinte ficheiro:

- 20191-team-17-ricardo/include/disciplina.hpp

Referência à classe Docente

```
#include <docente.hpp>
```

Diagrama de heranças da classe Docente

Membros públicos

- **Docente** (std::string nome, std::string cpf, std::string data_nasc, std::string endereco)
- virtual double **get_salario** ()=0
- virtual double **get_cart_trab** ()=0

Documentação dos Construtores & Destrutor

Docente::Docente (std::string *nome*, std::string *cpf*, std::string *data_nasc*, std::string *endereco*) [inline]

```
26                                     : Pessoa(nome, cpf, data_nasc, endereco)
{};
```

Documentação dos métodos

virtual double Docente::get_cart_trab () [pure virtual]

Implementado em **Professor** (p. 15).

virtual double Docente::get_salario () [pure virtual]

Implementado em **Professor** (p. 15).

A documentação para esta classe foi gerada a partir do seguinte ficheiro:

- 20191-team-17-ricardo/include/docente.hpp

Referência à classe Pessoa

```
#include <pessoa.hpp>
```

Diagrama de heranças da classe Pessoa

Membros públicos

- **Pessoa** (std::string nome, std::string cpf, std::string data_nasc, std::string endereco)
- std::string **get_nome** ()
- std::string **get_cpf** ()
- std::string **get_data_nasc** ()
- std::string **get_endereco** ()

Documentação dos Construtores & Destrutor

Pessoa::Pessoa (std::string *nome*, std::string *cpf*, std::string *data_nasc*, std::string *endereco*) [inline]

```
27
{
28         _nome = nome;
29         _cpf = cpf;
30         _data_nasc = data_nasc;
31         _endereco = endereco;
32     }
```

Documentação dos métodos

std::string Pessoa::get_cpf () [inline]

```
36         {
37             return _cpf;
38         };
```

std::string Pessoa::get_data_nasc () [inline]

```
39         {
40             return _data_nasc;
41         };
```

std::string Pessoa::get_endereco () [inline]

```
42         {
43             return _endereco;
44         };
```

std::string Pessoa::get_nome () [inline]

```
33         {
34             return _nome;
35         };
```

A documentação para esta classe foi gerada a partir do seguinte ficheiro:

- 20191-team-17-ricardo/include/pessoa.hpp

Referência à classe Professor

```
#include <professor.hpp>
```

Diagrama de heranças da classe Professor

Membros públicos

- **Professor** (std::string nome, std::string cpf, std::string data_nasc, std::string endereco, double salario, double cart_trab)
 - double **get_salario** ()
 - double **get_cart_trab** ()
-

Documentação dos Construtores & Destrutor

Professor::Professor (std::string *nome*, std::string *cpf*, std::string *data_nasc*, std::string *endereco*, double *salario*, double *cart_trab*)

```
14                                     : Docente(nome, cpf, data_nasc,
endereco){
15     _salario = salario;
16     _cart_trab = cart_trab;
17 }
```

Documentação dos métodos

double Professor::get_cart_trab () [virtual]

Implementa **Docente** (p. 13).

```
23                                     {
24     return _cart_trab;
25 }
```

double Professor::get_salario () [virtual]

Implementa **Docente** (p. 13).

```
19                                     {
20     return _salario;
21 }
```

A documentação para esta classe foi gerada a partir dos seguintes ficheiros:

- 20191-team-17-ricardo/include/**professor.hpp**
- 20191-team-17-ricardo/src/**professor.cpp**

Documentação do ficheiro

Referência ao ficheiro 20191-team-17-ricardo/include/aluno.hpp

Class **Aluno**.
`#include <iostream>`
`#include <iomanip>`
`#include <string>`
`#include "discente.hpp"`

Componentes

- class **Aluno**
-

Descrição detalhada

Class **Aluno**.

Autor:

Ricardo, Igor e Victor

Versão:

1.0

Referência ao ficheiro 20191-team-17-ricardo/include/departamento.hpp

Class **Departamento**.

```
#include <iostream>
```

```
#include <iomanip>
```

```
#include <string>
```

Componentes

- class **Departamento**

Descrição detalhada

Class **Departamento**.

Autor:

Ricardo, Igor e Victor

Versão:

1.0

Referência ao ficheiro 20191-team-17-ricardo/include/discente.hpp

Class **Discente**.

```
#include <iostream>
#include <iomanip>
#include <string>
#include "pessoa.hpp"
```

Componentes

- class **Discente**

Descrição detalhada

Class **Discente**.

Autor:

Ricardo, Igor e Victor

Versão:

1.0

Referência ao ficheiro 20191-team-17-ricardo/include/disciplina.hpp

Class **Disciplina**.

```
#include <iostream>
#include <iomanip>
#include <string>
```

Componentes

- class **Disciplina**

Descrição detalhada

Class **Disciplina**.

Autor:

Ricardo, Igor e Victor

Versão:

1.0

Referência ao ficheiro 20191-team-17-ricardo/include/docente.hpp

Class **Docente**.

```
#include <iostream>
#include <iomanip>
#include <string>
#include "pessoa.hpp"
```

Componentes

- class **Docente**

Descrição detalhada

Class **Docente**.

Autor:

Ricardo, Igor e Victor

Versão:

1.0

Referência ao ficheiro 20191-team-17-ricardo/include/pessoa.hpp

Class **Pessoa**.

```
#include <iostream>
#include <iomanip>
#include <string>
```

Componentes

- class **Pessoa**

Descrição detalhada

Class **Pessoa**.

Autor:

Ricardo, Igor e Victor

Versão:

1.0

Referência ao ficheiro 20191-team-17-ricardo/include/professor.hpp

Class **Professor**.

```
#include <iostream>
#include <iomanip>
#include <string>
#include "docente.hpp"
```

Componentes

- class **Professor**

Descrição detalhada

Class **Professor**.

Autor:

Ricardo, Igor e Victor

Versão:

1.0

Referência ao ficheiro 20191-team-17-ricardo/README.md

Referência ao ficheiro 20191-team-17-ricardo/src/aluno.cpp

```
#include <iostream>
#include <iomanip>
#include <string>
#include "aluno.hpp"
```


Referência ao ficheiro 20191-team-17-ricardo/src/departamento.cpp

```
#include <iostream>
#include <iomanip>
#include <string>
#include "departamento.hpp"
```

Referência ao ficheiro 20191-team-17-ricardo/src/disciplina.cpp

```
#include <iostream>
#include <iomanip>
#include <string>
#include "disciplina.hpp"
```

Referência ao ficheiro 20191-team-17-ricardo/src/main.cpp

```
#include <iostream>
#include <iomanip>
#include "pessoa.hpp"
#include "departamento.hpp"
#include "disciplina.hpp"
#include "professor.hpp"
#include "aluno.hpp"
#include <string>
#include <vector>
```

Funções

- `int main ()`

Documentação das funções

`int main ()`

```
14         {
15
16         double num_total_prof = 0;
17         double num_total_alun = 0;
18         double num_total_depa = 0;
19         double num_total_disc = 0;
20
21         std::string nome = "";
22         std::string cpf = "";
23         std::string data_nasc = "";
24         std::string endereco = "";
25         std::string semestre = "";
26         double matricula = 0;
27         double salario = 0;
28         double cart_trab = 0;
29         double num_alunos = 0;
30         double num_funcionarios = 0;
31
32         std::cout << "----- BEM VINDO -----" << std::endl;
33         std::cout << "" << std::endl;
34
35         std::cout << "Insira o número de professores a serem cadastrados: ";
36         std::cin >> num_total_prof;
37
38         std::cout << "Insira o número de alunos a serem cadastrados: ";
39         std::cin >> num_total_alun;
40
41         std::cout << "Insira o número de departamentos a serem cadastrados: ";
42         std::cin >> num_total_depa;
43
44         std::cout << "Insira o número de disciplinas a serem cadastradas: ";
45         std::cin >> num_total_disc;
46
47         std::vector<Professor> professores;
48         professores.reserve(num_total_prof);
49
50         for(int i = 0; i < num_total_prof; i++){
51
52             std::cout << "Professor " << i+1 << ":" << std::endl;
53
54             std::cout << "Nome: ";
55             std::cin >> nome;
56             std::cout << "" << std::endl;
57
58             std::cout << "CPF: ";
59             std::cin >> cpf;
60             std::cout << "" << std::endl;
61
62             std::cout << "Data Nascimento: ";
```

```

63         std::cin >> data_nasc;
64         std::cout << " " << std::endl;
65
66         std::cout << "Endereco: ";
67         std::cin >> endereco;
68         std::cout << " " << std::endl;
69
70         std::cout << "Salario: ";
71         std::cin >> salario;
72         std::cout << " " << std::endl;
73
74         std::cout << "Carteira de Trabalho: ";
75         std::cin >> cart_trab;
76         std::cout << " " << std::endl;
77
78         professores.push_back(Professor(nome, cpf, data_nasc, endereco,
salario, cart_trab));
79     }
80
81     std::vector<Aluno> alunos;
82     alunos.reserve(num_total_alun);
83
84     for(int i = 0; i < num_total_alun; i++){
85
86         std::cout << "Aluno " << i+1 << ":" << std::endl;
87
88         std::cout << "Nome: ";
89         std::cin >> nome;
90         std::cout << " " << std::endl;
91
92         std::cout << "CPF: ";
93         std::cin >> cpf;
94         std::cout << " " << std::endl;
95
96         std::cout << "Data Nascimento: ";
97         std::cin >> data_nasc;
98         std::cout << " " << std::endl;
99
100        std::cout << "Endereco: ";
101        std::cin >> endereco;
102        std::cout << " " << std::endl;
103
104        std::cout << "Matricula: ";
105        std::cin >> matricula;
106        std::cout << " " << std::endl;
107
108        alunos.push_back(Aluno(nome, cpf, data_nasc, endereco, matricula));
109    }
110
111    std::vector<Departamento> departamentos;
112    departamentos.reserve(num_total_depa);
113
114    for(int i = 0; i < num_total_depa; i++){
115
116        std::cout << "Departamento " << i+1 << ":" << std::endl;
117
118        std::cout << "Nome: ";
119        std::cin >> nome;
120        std::cout << " " << std::endl;
121
122        std::cout << "Numero de Funcionarios do Departamento: ";
123        std::cin >> num_funcionarios;
124        std::cout << " " << std::endl;
125
126        departamentos.push_back(Departamento(nome, num_funcionarios));
127    }
128
129    std::vector<Disciplina> disciplinas;
130    disciplinas.reserve(num_total_disc);
131
132    for(int i = 0; i < num_total_disc; i++){
133
134        std::cout << "Disciplina " << i+1 << ":" << std::endl;
135
136        std::cout << "Nome: ";
137        std::cin >> nome;
138        std::cout << " " << std::endl;

```

```

139
140         std::cout << "Semestre: ";
141         std::cin >> semestre;
142         std::cout << " " << std::endl;
143
144         std::cout << "Numero de Alunos Matriculados: ";
145         std::cin >> num_alunos;
146         std::cout << " " << std::endl;
147
148         disciplinas.push_back(Disciplina(nome, semestre, num_alunos));
149     }
150
151
152     std::cout << " " << std::endl;
153     std::cout <<
    "-----" << std::endl;
154     std::cout << " " << std::endl;
155
156     for(int i = 0; i < num_total_prof; i++){
157
158         std::cout << "Professor " << i+1 << ":" << std::endl;
159
160         std::cout << "Nome: " << professores[i].get_nome() << std::endl;
161         std::cout << "CPF: " << professores[i].get_cpf() << std::endl;
162         std::cout << "Data Nascimento: " << professores[i].get_data_nasc()
<< std::endl;
163         std::cout << "Endereco: " << professores[i].get_endereco() <<
std::endl;
164         std::cout << "Salario: " << professores[i].get_salario() <<
std::endl;
165         std::cout << "Carteira de Trabalho: " <<
professores[i].get_cart_trab() << std::endl;
166         std::cout << " " << std::endl;
167     }
168
169     std::cout << " " << std::endl;
170     std::cout <<
    "-----" << std::endl;
171     std::cout << " " << std::endl;
172
173     for(int i = 0; i < num_total_alun; i++){
174
175         std::cout << "Aluno " << i+1 << ":" << std::endl;
176
177         std::cout << "Nome: " << alunos[i].get_nome() << std::endl;
178         std::cout << "CPF: " << alunos[i].get_cpf() << std::endl;
179         std::cout << "Data Nascimento: " << alunos[i].get_data_nasc() <<
std::endl;
180         std::cout << "Endereco: " << alunos[i].get_endereco() << std::endl;
181         std::cout << "Matricula: " << alunos[i].get_matricula() <<
std::endl;
182         std::cout << " " << std::endl;
183     }
184
185     std::cout << " " << std::endl;
186     std::cout <<
    "-----" << std::endl;
187     std::cout << " " << std::endl;
188
189     for(int i = 0; i < num_total_depa; i++){
190
191         std::cout << "Departamento " << i+1 << ":" << std::endl;
192
193         std::cout << "Nome: " << departamentos[i].get_nome() << std::endl;
194         std::cout << "Num Funcionarios: " <<
departamentos[i].get_num_funcionarios() << std::endl;
195         std::cout << " " << std::endl;
196     }
197
198     std::cout << " " << std::endl;
199     std::cout <<
    "-----" << std::endl;
200     std::cout << " " << std::endl;
201
202     for(int i = 0; i < num_total_disc; i++){
203
204         std::cout << "Disciplina " << i+1 << ":" << std::endl;

```

```

205
206         std::cout << "Nome: " << disciplinas[i].get_nome() << std::endl;
207         std::cout << "Semestre: " << disciplinas[i].get_semestre() <<
std::endl;
208         std::cout << "Numero de Alunos: " << disciplinas[i].get_num_alunos()
<< std::endl;
209         std::cout << "" << std::endl;
210     }
211
212     std::cout << "" << std::endl;
213     std::cout <<
"-----" << std::endl;
214     std::cout << "" << std::endl;
215
216     return 0;
217 }

```

Referência ao ficheiro 20191-team-17-ricardo/src/professor.cpp

```
#include <iostream>
#include <iomanip>
#include <string>
#include "professor.hpp"
```

Referência ao ficheiro 20191-team-17-ricardo/user_stories.md

