Projeto Final - Linguagem de Programação I
Sistema de Folha Salarial

ARTHUR HENRIQUE DA SILVA - 20200077587
AVANI MARIA DA FONSECA - 20210067000
ISAAC SEBASTIAN LIMA DE ARAUJO - 20210025403
Junho de 2022

Sumário

Índice Hierárquico

Hierarquia de Classes

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

Índice dos Componentes

Lista de Classes

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

Índice dos Arquivos

Lista de Arquivos

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

Classes

Referência da Classe Data

`#include <Data.h>`

Membros Públicos

Data ()
Construtor 1.

Data (int d, int m, int a)
Construtor 2.

Data (string txt)
Construtor 3.

virtual **~Data** ()
Destrutor.

int **getDia** ()

int **getMes** ()

int **getAano** ()

void **setDia** (int d)

void **setMes** (int m)

void **setAno** (int a)

int **getQuantDiasDoMes** (int m)

int **getDiaDoAno** ()

string **to_txt** ()
Prepara a classe para ser escrita em arquivos.

Construtores e Destrutores

Data::Data ()

Construtor 1.
Construtor vazio.

Data::Data (int *d*, int *m*, int *a*)

Construtor 2.

Recebe os atributos dia, mês e ano.

Parâmetros

d	dia
m	mês
a	ano

Data::Data (string *txt*)

Construtor 3.

Recebe uma string com dia, mês e ano separados por barras

Data::~Data ()[virtual]

Destrutor.

Funções membros

int Data::getDia Ano ()

Retorna

a posição da data em relação aos 365 dias do ano (int).

int Data::getQuantDiasDoMes (int *m*)

Parâmetros

m	mês

Retorna

a quantidade de dias do mês (int).

string Data::to_txt ()

Prepara a classe para ser escrita em arquivos.

Retorna

string com atributos dia, mês e ano separados por barras

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

Data.h
Data.cpp

Referência da Classe Diretor

`#include <Diretor.h>`

Membros Públicos

Diretor ()
Construtor padrão.

Diretor (std::string areaSupervisao, std::string areaFormacao)
Construtor com strings.

Diretor (std::string txt)
virtual **~Diretor** ()
Destrutor.

std::string **getAreasupervisao** ()
getAreasupervisao.

std::string **getAreaformacao** ()
getAreaformacao.

void **setAreasupervisao** (std::string areaSupervisao)
setAreasupervisao.

void **setAreaformacao** (std::string areaFormacao)
setAreaformacao.

std::string **to_txt** ()
void **aumentaSalario** ()
aumentaSalario.

Outros membros herdados

Construtores e Destrutores

Diretor::Diretor ()

Construtor padrão.

Diretor::Diretor (std::string *areaSupervisao*, std::string *areaFormacao*)

Construtor com strings.

Construtor que atribui uma string ao atributo areaSupervisao e outra ao areaFormacao.

Parâmetros

areaSupervisao	string que armazena a área de supervisão do funcionário.
areaFormacao	string que armazena a área de formação do funcionário.

Diretor::Diretor (std::string *txt*)

Diretor::~Diretor ()[virtual]

Destrutor.

Funções membros

void Diretor::aumentaSalario ()[virtual]

aumentaSalario.

Método herdado que aumenta o atributo salário em 20%.

Retorna

void.

Reimplementa **Funcionario** (*p.*).

std::string Diretor::to_txt ()[virtual]

Reimplementa **Funcionario** (*p.*).

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

Diretor.h
Diretor.cpp

Referência da Classe Endereco

`#include <Endereco.h>`

Membros Públicos

Endereco ()
Construtor 1.

Endereco (string cep)
Construtor 2.

Endereco (string txt, int i)
Construtor 3.

virtual **~Endereco** ()
Destrutor.

string **to_txt** ()

void **print** ()

string **getRua** ()

string **getBairro** ()

string **getCidade** ()

string **getEstado** ()

void **setRua** (string rua)

void **setBairro** (string bairro)

void **setCidade** (string cidade)

void **setEstado** (string estado)

Construtores e Destrutores

Endereco::Endereco ()

Construtor 1.
Construtor vazio

Endereco::Endereco (string *cep*)

Construtor 2.
Utiliza cep entregue como parâmetro para obter informação pela API viacep

Parâmetros

string	Cep do usuario

Endereco::Endereco (string *txt*, int *i*)

Construtor 3.

Parâmetros

string	Uma string com os atributos Rua, Bairro, Cidade e Estado separados por ponto e vírgula.
int	Diferencia o Construtor 1 e o Construtor 2

Endereco::~Endereco ()[virtual]

Destrutor.

Funções membros

void Endereco::print ()

Exibe os atributos rua, bairro, cidade e estado no console

string Endereco::to_txt ()

Retorna

string com atributos Rua, Bairro, Cidade e Estado separados por ponto e vírgula

---

Atributos

int Data::dia[protected]

Dia da data..

int Data::mes[protected]

Mês da data..

int Data::ano[protected]

Ano da data..

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

Endereco.h
Endereco.cpp

Referência da Classe Funcionario

`#include <Funcionario.h>`

Membros Públicos

Funcionario ()
Construtor padrão.

Funcionario (string txt)
virtual **~Funcionario** ()
Destrutor.

void **setFuncionario** (std::string **codigo**, std::string **nome**, std::string cep, std::string **telefone**, int designacaoN, int dia, int mes, int ano)
setFuncionario.

std::string **getCodigo** ()
getCodigo

std::string **getNome** ()
getNome

Endereco getEndereco ()
getEndereco

std::string **getTelefone** ()
getTelefone

Data getDataAdmissao ()
getDataAdmissao

std::string **getDesignacao** ()
getDesignacao

double **getSalario** ()
getTelefone

void **setCodigo** (std::string **codigo**)
setCodigo

void **setNome** (std::string **nome**)
setNome

void **setEndereco** (std::string cep)

setEndereco

void **setTelefone** (std::string **telefone**)
setTelefone

void **setDataAdmissao** (int dia, int mes, int ano)
setDataAdmissao

void **setDesignacao** (int designacaoN)
setDesignacao

void **setSalario** (double **salario**)
setSalario

virtual string **to_txt** ()
virtual void **aumentaSalario** ()
aumentaSalario

Membros Protegidos

string **Clean_CSV** (string &str, string find, int reads)

Atributos Protegidos

std::string **codigo**
std::string **nome**
Endereco endereco
std::string **telefone**
Data dataAdmissao
std::string **designacao**
double **salario**

Construtores e Destrutores

Funcionario::Funcionario ()

Construtor padrão.

Funcionario::Funcionario (string *txt*)

Funcionario::~Funcionario ()[virtual]

Destrutor.

Funções membros

void Funcionario::aumentaSalario ()[virtual]

aumentaSalario

Método herdado pelas classes filhas, aumentando o atributo salario em diferentes porcentagens.

Retorna

void.

Reimplementado por **Presidente** (*p.*), **Operador** (*p.*), **Gerente** (*p.*) e **Diretor** (*p.*).

string Funcionario::Clean_CSV (string & *str*, string *find*, int *reads*)[protected]

void Funcionario::setFuncionario (std::string *codigo*, std::string *nome*, std::string *cep*, std::string *telefone*, int *designacaoN*, int *dia*, int *mes*, int *ano*)

setFuncionario.

Atribui as informações do funcionário aos atributos da classe.

Parâmetros

codigo	uma string que armazena o código de identificação do funcionário.
nome	uma string que armazena o nome completo do funcionário.
cep	uma string que armazena o cep do funcionário.
telefone	uma string que armazena o telefone para contato do funcionário.
designacaoN	inteiro usado para indicar a ocupação do funcionário na empresa.
dia	inteiro que armazena o dia de admissão.
mes	inteiro que armazena o mês de admissão.
ano	inteiro que armazena o ano de admissão.

Retorna

void

Veja também

setEndereco(), setDataadmissao() e **setDesignacao()**

string Funcionario::to_txt ()[virtual]

Reimplementado por **Presidente** (*p.*), **Gerente** (*p.*) e **Diretor** (*p.*).

Atributos

std::string Funcionario::codigo[protected]

Código do funcionário.

Data Funcionario::dataAdmissao[protected]

Composição da classe **Data**, armazenando a data de admissão do funcionário.

std::string Funcionario::designacao[protected]

Indica se o funcionário é um operador, gerente, diretor ou presidente.

Endereco Funcionario::endereco[protected]

Composição da classe **Endereco**, armazenando informações extraídas do CEP.

std::string Funcionario::nome[protected]

Nome completo do funcionário.

double Funcionario::salario[protected]

Armazena o salário do funcionário.

std::string Funcionario::telefone[protected]

Telefone do funcionário.

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

Funcionario.h
Funcionario.cpp

Referência da Classe Gerente

`#include <Gerente.h>`

Membros Públicos

Gerente ()
Construtor padrão.

Gerente (std::string areaSupervisao)
Construtor com string.

Gerente (std::string txt, int i)
virtual **~Gerente** ()
Destrutor.

std::string **getAreasupervisao** ()
getAreasupervisao.

void **setAreasupervisao** (std::string areaSupervisao)
setAreasupervisao.

void **aumentaSalario** ()
aumentaSalario.

std::string **to_txt** ()

Outros membros herdados

Construtores e Destrutores

Gerente::Gerente ()

Construtor padrão.

Gerente::Gerente (std::string *areaSupervisao*)

Construtor com string.

Construtor que atribui uma string ao atributo areaSupervisao.

Parâmetros

areaSupervisao	string que armazena a área de supervisão do funcionário.

Gerente::Gerente (std::string *txt*, int *i*)

Gerente::~Gerente ()[virtual]

Destrutor.

Funções membros

void Gerente::aumentaSalario ()[virtual]

aumentaSalario.

Método herdado que aumenta o atributo salário em 10%.

Retorna

void.

Reimplementa **Funcionario** (*p.*).

std::string Gerente::to_txt ()[virtual]

Reimplementa **Funcionario** (*p.*).

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

Gerente.h
Gerente.cpp

Referência da Classe Operador

`#include <Operador.h>`

Membros Públicos

Operador ()
Construtor padrão.

Operador (string txt)
~Operador ()
Destrutor.

void **aumentaSalario** ()
aumentaSalario.

Construtores e Destrutores

Operador::Operador ()

Construtor padrão.

Operador::Operador (string *txt*)

Recebe os atributos separados por ponto e vírgula.

Operador::~Operador ()

Destrutor.

Funções membros

void Operador::aumentaSalario ()[virtual]

aumentaSalario.
Método herdado que aumenta o atributo salario em 5%.

Retorna
void.

Reimplementa **Funcionario** (*p.*).

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

- **Operador.h**
- **Operador.cpp**

Referência da Classe Presidente

`#include <Presidente.h>`

Membros Públicos

Presidente ()
Construtor padrão.

Presidente (std::string formacaoAcademica, std::string areaFormacao)
Construtor com strings.

Presidente (string txt)
virtual **~Presidente** ()
Destrutor.

std::string **getFormacaoacademica** ()
getFormacaoacademica.

std::string **getAreaformacao** ()
getAreaformacao.

void **setFormacaoacademica** (std::string formacaoAcademica)
setFormacaoacademica.

void **setAreaformacao** (std::string areaFormacao)
setAreaformacao.

std::string **to_txt** ()
void **aumentaSalario** ()
aumentaSalario.

Construtores e Destrutores

Presidente::Presidente ()

Construtor padrão.

Presidente::Presidente (std::string *formacaoAcademica*, std::string *areaFormacao*)

Construtor com strings.

Construtor que atribui uma string ao atributo areaFormacao e outra ao formacaoAcademica.

Parâmetros

areaFormacao	string que armazena a área de formação do funcionário.

formacaoAcademica	string que armazena o nível de formação acadêmica do funcionário.

Presidente::Presidente (string *txt*)

Recebe os atributos separados por ponto e vírgula.

Presidente::~Presidente ()[virtual]

Destrutor.

Funções membros

void Presidente::aumentaSalario ()[virtual]

aumentaSalario.

Método herdado que aumenta o atributo salário em 30%.

Retorna

void.

Reimplementa **Funcionario** (*p.*).

std::string Presidente::to_txt ()[virtual]

Reimplementa **Funcionario** (*p.*).

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

Presidente.h
Presidente.cpp

Referência da Classe SistemaGerenciaFuncionario

`#include <SistemaGerenciaFuncionario.h>`

Membros Públicos

SistemaGerenciaFuncionario ()
Construtor padrão.

virtual **~SistemaGerenciaFuncionario** ()
Destrutor.

void **pushOperador** ()
pushOperador.

void **pushGerente** ()
pushGerente.

void **pushDiretor** ()
pushDiretor.

void **pushPresidente** ()
pushPresidente.

void **setFuncionario** (std::string cod, std::string nom, std::string ende, std::string tel, int desig, int d, int m, int a, int pos)
setFuncionario.

void **setAreasupervisao** (std::string areaS, int pos)
setAreasupervisao.

void **setAreaformacao** (std::string areaF, int pos)
setAreaformacao.

void **setFormacaoacademica** (std::string formacaoA, int pos)
setFormacaoacademica.

void **setCodigo** (std::string codigo, int pos)
setCodigo.

void **setNome** (std::string nome, int pos)
setNome.

void **setEndereco** (std::string endereco, int pos)
setEndereco.

void **setTelefone** (std::string telefone, int pos)
setTelefone.

void **setDesignacao** (int designacaoN, int pos)
setDesignacao.

void **setSalario** (double salario, int pos)
setSalario.

void **setDataAdmissao** (int dia, int mes, int ano, int pos)
setDataAdmissao.

int **getSize** ()
getSize.

void **imprimeFuncionario** (int pos)
imprimeFuncionario.

void **imprimeFuncionarioDesig** (std::string desig, int pos)
imprimeFuncionarioDesig.

void **deletaFuncionario** (int pos)
deletaFuncionario.

int **buscaCodFuncionario** (std::string cod)
buscaCodFuncionario.

void **buscaNomeFuncionario** (std::string nome)
buscaNomeFuncionario.

void **buscaDataDia** (int d1, int d2, int m, int a)
buscaDataDia.

void **buscaDataMes** (int m1, int m2, int a)
buscaDataMes.

void **buscaDataAno** (int a1, int a2)
buscaDataMes.

int **buscaNomeFuncionarioIndex** (std::string nome)
buscaNomeFuncionarioIndex.

std::string **getDesigFuncionario** (int pos)
setSalario.

int **existeDesigFuncionario** (std::string desig)
existeDesigFuncionario.

void **aumentoDeSalarios** ()
aumentoDeSalario.

double **folhaSalarialFuncionario** (int pos)
folhaSalarialFuncionario.

void **folhaSalarialAnual** (int ano)
folhaSalarialAnual.

void **buscaEnderecoRua** (string _rua)
buscaEnderecoRua.

void **buscaEnderecoBairro** (string _bairro)
buscaEnderecoBairro.

void **buscaEnderecoCidade** (string _cidade)
buscaEnderecoCidade.

void **buscaEnderecoEstado** (string _estado)
buscaEnderecoEstado.

void **geraFolhaMensal** (int mes, int ano, bool anual=false)
geraFolhaMensal.

void **imprimeFolha** (int mes, int ano)
imprimeFolha.

Construtores e Destrutores

SistemaGerenciaFuncionario::SistemaGerenciaFuncionario ()

Construtor padrão.

SistemaGerenciaFuncionario::~SistemaGerenciaFuncionario ()[virtual]

Destrutor.

Funções membros

void SistemaGerenciaFuncionario::aumentoDeSalarios ()

aumentoDeSalario.

Método que chama o respectivo aumentaSalario() de cada objeto do vector.

Retorna

void.

int SistemaGerenciaFuncionario::buscaCodFuncionario (std::string *cod*)

buscaCodFuncionario.

Faz uma busca do código passado dentre os já cadastrados e retorna, caso encontrado, seu índice no vector.

Parâmetros

cod	string que armazena o código a ser buscado.

Retorna

Um inteiro.

void SistemaGerenciaFuncionario::buscaDataAno (int *a1*, int *a2*)

buscaDataMes.

Faz uma busca dos funcionários admitidos entre dois anos, imprimindo, caso encontrado, os seus dados.

Parâmetros

a1	inteiro que armazena o ano no qual a busca se iniciará.
a2	inteiro que armazena o ano no qual a busca se encerrará.

Retorna

void.

void SistemaGerenciaFuncionario::buscaDataDia (int *d1*, int *d2*, int *m*, int *a*)

buscaDataDia.

Faz uma busca dos funcionários admitidos entre dois dias dentro de um mesmo mês e ano, imprimindo, caso encontrado, os seus dados.

Parâmetros

d1	inteiro que armazena o dia no qual a busca se iniciará.
d2	inteiro que armazena o dia no qual a busca se encerrará.
m	inteiro que indica o mês no qual será realizada a busca.
a	inteiro que indica o ano no qual será realizada a busca.

Retorna

void.

void SistemaGerenciaFuncionario::buscaDataMes (int *m1*, int *m2*, int *a*)

buscaDataMes.

Faz uma busca dos funcionários admitidos entre dois meses dentro de um mesmo ano, imprimindo, caso encontrado, os seus dados.

Parâmetros

m1	inteiro que armazena o mês no qual a busca se iniciará.
m2	inteiro que armazena o mês no qual a busca se encerrará.
a	inteiro que indica o ano no qual será realizada a busca.

Retorna

void.

void SistemaGerenciaFuncionario::buscaEnderecoBairro (string *_bairro*)

buscaEnderecoBairro.

Busca por funcionários que residem em um determinado bairro.

Parâmetros

_bairro	string que guarda o bairro usado na busca.

Retorna

void.

void SistemaGerenciaFuncionario::buscaEnderecoCidade (string *_cidade*)

buscaEnderecoCidade.

Busca por funcionários que residem em uma determinada cidade.

Parâmetros

_cidade	string que guarda a cidade usada na busca.

Retorna

void.

void SistemaGerenciaFuncionario::buscaEnderecoEstado (string *_estado*)

buscaEnderecoEstado.

Busca por funcionários que residem em um determinado estado.

Parâmetros

_estado	string que guarda a sigla do estado usado na busca.

Retorna

void.

void SistemaGerenciaFuncionario::buscaEnderecoRua (string *_rua*)

buscaEnderecoRua.

Busca por funcionários que residem em uma determinada rua.

Parâmetros

_rua	string que guarda a rua usada na busca.

Retorna

void.

void SistemaGerenciaFuncionario::buscaNomeFuncionario (std::string *nome*)

buscaNomeFuncionario.

Faz uma busca do nome passado dentre os já cadastrados e imprime, caso encontrado, seus dados.

Parâmetros

nome	string que armazena o nome a ser buscado.

Retorna

void.

int SistemaGerenciaFuncionario::buscaNomeFuncionarioIndex (std::string *nome*)

buscaNomeFuncionarioIndex.

Faz uma busca do nome passado dentre os já cadastrados e retorna, caso encontrado, seu índice no vector.

Parâmetros

nome	string que armazena o nome a ser buscado.

Retorna

Um inteiro.

void SistemaGerenciaFuncionario::deletaFuncionario (int *pos*)

deletaFuncionario.

Método que remove do vector funcionarios o objeto armazenado no índice passado.

Parâmetros

pos	inteiro que indica o índice do vector ao qual as informações serão atribuídas.

Retorna

void.

int SistemaGerenciaFuncionario::existeDesigFuncionario (std::string *desig*)

existeDesigFuncionario.

Faz uma busca da designação passada dentre os já cadastrados e retorna, caso encontrado, um valor positivo.

Parâmetros

desig	string que armazena a designação a ser buscada.

Retorna

Um inteiro.

void SistemaGerenciaFuncionario::folhaSalarialAnual (int *ano*)

folhaSalarialAnual.

Exibe os gastos mensais da empresa e o total anual.

Parâmetros

ano	ano de referência.

Retorna

void.

double SistemaGerenciaFuncionario::folhaSalarialFuncionario (int *pos*)

folhaSalarialFuncionario.

Método que calcula, imprime e retorna o salário líquido do funcionário.

Parâmetros

pos	inteiro que indica o índice do funcionário que terá seu salário líquido calculado.

Retorna

Um double.

Veja também

void SistemaGerenciaFuncionario::geraFolhaMensal (int *mes*, int *ano*, bool *anual* = `false`)

geraFolhaMensal.

Método que calcula e imprime o salário líquido dos funcionários e total pago pela empresa dentro de um mês.

Parâmetros

mes	mes de referência.
ano	ano de referência.
bool-anual	indica que mais de uma folha do mesmo ano está sendo gerada ao mesmo tempo.

Retorna

void.

std::string SistemaGerenciaFuncionario::getDesigFuncionario (int *pos*)

setSalario.

Método que chama o getDesignacao() do objeto de tipo **Funcionario**.

Parâmetros

pos	inteiro que indica o índice do vector do qual as informações serão extraídas.

Retorna

Uma string.

int SistemaGerenciaFuncionario::getSize ()

getSize.

Método que retorna a quantidade de objetos armazenados no vector funcionarios (o seu tamanho).

Retorna

Um inteiro.

void SistemaGerenciaFuncionario::imprimeFolha (int *mes*, int *ano*)

imprimeFolha.

Abre arquivo e exibe suas informações.

Parâmetros

mes	mes de referência.
ano	ano de referência.

Retorna

void.

void SistemaGerenciaFuncionario::imprimeFuncionario (int *pos*)

imprimeFuncionario.

Método que imprime os dados do objeto **Funcionario** encontrado no índice passado.

Parâmetros

pos	inteiro que indica o índice do vector ao qual as informações serão atribuídas.

Retorna

void.

void SistemaGerenciaFuncionario::imprimeFuncionarioDesig (std::string *desig*, int *pos*)

imprimeFuncionarioDesig.

Método que imprime os dados do objeto **Funcionario** encontrado no índice passado caso coincida com a designação passada.

Parâmetros

desig	string que indica quais funcionários devem ser impressos de acordo com sua ocupação.
pos	inteiro que indica o índice do vector ao qual as informações serão atribuídas.

Retorna

void.

void SistemaGerenciaFuncionario::pushDiretor ()

pushDiretor.

Aloca no vector espaco na memória para um objeto do tipo **Diretor**.

Retorna

void.

void SistemaGerenciaFuncionario::pushGerente ()

pushGerente.

Aloca, no vector, espaço na memória para um objeto do tipo **Gerente**.

Retorna

void.

void SistemaGerenciaFuncionario::pushOperador ()

pushOperador.

Aloca, no vector, espaço na memória para um objeto do tipo **Operador**.

Retorna

void.

void SistemaGerenciaFuncionario::pushPresidente ()

pushPresidente.

Aloca, no vector, espaço na memória para um objeto do tipo **Presidente**.

Retorna

void.

void SistemaGerenciaFuncionario::setFuncionario (std::string *cod*, std::string *nom*, std::string *ende*, std::string *tel*, int *desig*, int *d*, int *m*, int *a*, int *pos*)

setFuncionario.

Chama os métodos set da classe **Funcionario** e atribui as informações do funcionário no índice do vector.

Parâmetros

cod	uma string que armazena o código de identificação do funcionário.
nom	uma string que armazena o nome completo do funcionário.
ende	uma string que armazena o cep do funcionário.
tel	uma string que armazena o telefone para contato do funcionário.
desig	inteiro usado para indicar a ocupação do funcionário na empresa.
d	inteiro que armazena o dia de admissão.
m	inteiro que armazena o mês de admissão.
a	inteiro que armazena o ano de admissão.
pos	inteiro que indica o índice do vector ao qual as informações serão atribuídas.

Retorna

void

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

SistemaGerenciaFuncionario.h
SistemaGerenciaFuncionario.cpp

Arquivos

Referência do Arquivo Data.cpp

```
#include "Data.h"
```

Referência do Arquivo Data.h

```
#include <iostream>
```

Componentes

class **Data**

Referência do Arquivo Diretor.cpp

```
#include "Diretor.h"
```

Definições e Macros

#define **AUMENTO_BASE 20**

Definições e macros

#define AUMENTO_BASE 20

Valor:

Referência do Arquivo Diretor.h

```
#include "Funcionario.h"
```

Componentes

class **Diretor**

Referência do Arquivo Endereco.cpp

```
#include "Endereco.h"
```

Referência do Arquivo Endereco.h

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

Componentes

class **Endereco**

Referência do Arquivo Funcionario.cpp

```
#include "Funcionario.h"
```

Definições e Macros

#define **SALARIO_BASE** 1320

Definições e macros

#define SALARIO_BASE 1320

Referência do Arquivo Funcionario.h

```
#include <iostream>
#include <string>
#include "Data.h"
#include "Endereco.h"
```

Componentes

class **Funcionario**

Referência do Arquivo Gerente.cpp

```
#include "Gerente.h"
```

Definições e Macros

#define **AUMENTO_BASE 10**

Definições e macros

#define AUMENTO_BASE 10

Referência do Arquivo Gerente.h

```
#include "Funcionario.h"
```

Componentes

class **Gerente**

Referência do Arquivo main.cpp

```
#include <stdlib.h>
#include "SistemaGerenciaFuncionario.h"
```

Funções

void **mySleep** (int sleepMs)
int **main** ()
void **caracteres** ()

Funções

int main ()

void mySleep (int *sleepMs*)

void caracteres ()

Referência do Arquivo Operador.cpp

```
#include "Operador.h"
```

Definições e Macros

#define **AUMENTO_BASE** 5

Definições e macros

#define AUMENTO_BASE 5

Referência do Arquivo Operador.h

```
#include "Funcionario.h"
```

Componentes

class **Operador**

Referência do Arquivo Presidente.cpp

`#include "Presidente.h"`

Definições e Macros

#define **AUMENTO_BASE 30**

Definições e macros

#define AUMENTO_BASE 30

Referência do Arquivo Presidente.h

`#include "Funcionario.h"`

Componentes

class **Presidente**

Referência do Arquivo SistemaGerenciaFuncionario.cpp

```
#include "SistemaGerenciaFuncionario.h"
```

Referência do Arquivo SistemaGerenciaFuncionario.h

```
#include <ctime>
#include <vector>
#include <fstream>
#include <string>
#include "Data.h"
#include "Funcionario.h"
#include "Diretor.h"
#include "Gerente.h"
#include "Operador.h"
#include "Presidente.h"
```

Componentes

class **SistemaGerenciaFuncionario**