

Sistema de cadastro e manipulação de alarmes relacionados a equipamentos

V0.1

Gerado por Doxygen 1.8.17

1 Sistema de cadastro e manipulação de alarmes relacionados a equipamentos	1
1.1 Resumo	1
2 Lista de atividades futuras	3
3 Índice das Estruturas de Dados	5
3.1 Estruturas de Dados	5
4 Índice dos Arquivos	7
4.1 Lista de Arquivos	7
5 Estruturas	9
5.1 Referência da Estrutura Alarme_t	9
5.1.1 Descrição detalhada	9
5.1.2 Campos	9
5.1.2.1 Ativo	9
5.1.2.2 Classificacao	10
5.1.2.3 DataCadastro	10
5.1.2.4 DataEntrada	10
5.1.2.5 DataSaida	10
5.1.2.6 Descricao	10
5.1.2.7 identificador	10
5.1.2.8 N_Ativacao	10
5.1.2.9 N_Serie_Equi	10
5.1.2.10 Nome	11
5.2 Referência da Estrutura Data_t	11
5.2.1 Descrição detalhada	11
5.2.2 Campos	11
5.2.2.1 Ano	11
5.2.2.2 Dia	11
5.2.2.3 Mes	12
5.3 Referência da Estrutura Equipamento_t	12
5.3.1 Descrição detalhada	12
5.3.2 Campos	12
5.3.2.1 Data	12
5.3.2.2 Descricao	12
5.3.2.3 N_Serie	13
5.3.2.4 Nome	13
5.3.2.5 Tipo	13
5.4 Referência da Estrutura IntAlarm_t	13
5.4.1 Descrição detalhada	13
5.4.2 Campos	13
5.4.2.1 Id	14
5.4.2.2 Val	14

5.5 Referência da Estrutura StrAlarm_t	14
5.5.1 Descrição detalhada	14
5.5.2 Campos	14
5.5.2.1 Desc	14
5.5.2.2 Id	14
6 Arquivos	15
6.1 Referência do Arquivo Alarme.h	15
6.1.1 Descrição detalhada	15
6.1.2 Funções	16
6.1.2.1 CreateAlarme()	16
6.1.2.2 DeleteAlarme()	16
6.1.2.3 ReadAlarme()	16
6.1.2.4 String2Alarme_t()	17
6.1.2.5 UpdateAlarme()	17
6.2 Referência do Arquivo Doc_pages/mainpage.md	18
6.3 Referência do Arquivo Equipamento.h	18
6.3.1 Descrição detalhada	18
6.3.2 Definições dos tipos	19
6.3.2.1 Equipamento_t	19
6.3.3 Funções	19
6.3.3.1 CreateEquipamento()	19
6.3.3.2 DeleteEquipamento()	19
6.3.3.3 GetInt()	20
6.3.3.4 Limpar_buffer()	20
6.3.3.5 ReadEquipamento()	20
6.3.3.6 String2Equipamento_t()	21
6.3.3.7 UpdateEquipamento()	21
6.4 Referência do Arquivo Interface.h	21
6.4.1 Descrição detalhada	23
6.4.2 Funções	23
6.4.2.1 Ativar_Alarmes()	23
6.4.2.2 Atualiza_Alarme()	23
6.4.2.3 Atualiza_Equipamento()	23
6.4.2.4 Criar_Alarme()	24
6.4.2.5 Criar_Equipamento()	24
6.4.2.6 Desativar_Alarmes()	24
6.4.2.7 Filtrar()	24
6.4.2.8 GetSize()	24
6.4.2.9 ImprimirAlarme()	25
6.4.2.10 ImprimirAtuado()	25
6.4.2.11 ImprimirEquipamento()	25

6.4.2.12 ImprimirTodosAtuados()	25
6.4.2.13 ImprimirTudoAlarmes()	26
6.4.2.14 ImprimirTudoEquipamentos()	26
6.4.2.15 Menu()	26
6.4.2.16 Ordenar_Desc()	26
6.4.2.17 OrdenarAtuacao()	26
6.4.2.18 OrdenarClassificacao()	26
6.4.2.19 Pesquisa_Equipamento()	27
6.4.2.20 Pesquisar_Alarme()	27
6.4.2.21 Remover_Alarme()	27
6.4.2.22 Remover_Equipamento()	27
6.4.2.23 SortInt()	27
6.4.2.24 SortString()	28
6.5 Referência do Arquivo Sistema.c	28
6.5.1 Descrição detalhada	28
6.5.2 Funções	29
6.5.2.1 main()	29
6.6 Referência do Arquivo Sistema.h	29
6.6.1 Descrição detalhada	30
6.6.2 Definições e macros	30
6.6.2.1 ALARME_FILE	30
6.6.2.2 BYTES_MAX	30
6.6.2.3 DESC_MAX	31
6.6.2.4 EQUIP_FILE	31
6.6.2.5 FALSE	31
6.6.2.6 INT_MAX	31
6.6.2.7 NOME_MAX	31
6.6.2.8 TEMP	31
6.6.2.9 TRUE	32
6.6.3 Definições dos tipos	32
6.6.3.1 bool	32
6.6.3.2 Data_t	32
6.6.3.3 IntAlarm_t	32
6.6.3.4 StrAlarm_t	32
6.6.4 Enumerações	32
6.6.4.1 anonymous enum	32
6.6.4.2 anonymous enum	33
Índice Remissivo	35

Capítulo 1

Sistema de cadastro e manipulação de alarmes relacionados a equipamentos

1.1 Resumo

Sistema que tem como objetivo a manipulação e cadastro de alarmes relacionados a equipamentos, foi desenvolvido utilizando dois arquivos do tipo .tsv que podem ser abertos como tabela. Esses dois arquivos são manipulados para a inserção/remoção/atualização/pesquisa de elementos como alarmes e equipamentos.

O sistema foi desenvolvido com essas características com o intuito de economizar memória, para assim não ser necessário armazenar todos os dados em listas encadeadas na memória durante a execução do sistema e só armazenar os dados ao fim da execução.

O sistema possui 5 arquivos:

1. **Sistema.c**: Arquivo contendo a função main e todos os includes.
2. **Sistema.h**: Arquivo contendo as definições e tipos de dados utilizados no sistema.
3. **Interface.h**: Arquivo contendo as funções de interface e suas funções auxiliares.
4. **Equipamento.h**: Arquivo contendo o tipo de dado `Equipamento_t`, suas funções de CRUD as suas funções auxiliares.
5. **Alarme.h**: Arquivo contendo o tipo de dado `Alarme_t`, as suas funções de CRUD e suas funções auxiliares.

Capítulo 2

Lista de atividades futuras

Global [Data_t](#)

Utilizar a time.h para definir datas.

Capítulo 3

Índice das Estruturas de Dados

3.1 Estruturas de Dados

Aqui estão as estruturas de dados, uniões e suas respectivas descrições:

Alarme_t	Estrutura de dados para o armazenamento e manipulação de alarmes	9
Data_t	Tipo de dados para datas	11
Equipamento_t	Estrutura de dados para o armazenamento e manipulação de equipamentos	12
IntAlarm_t	Estrutura de dados que contém um identificador de alarme e um inteiro, utilizada para a ordenação dos alarmes	13
StrAlarm_t	Estrutura de dados que contém um identificador de alarme e uma string, utilizado para ordenação de strings	14

Capítulo 4

Índice dos Arquivos

4.1 Lista de Arquivos

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

Alarme.h	Arquivo contendo o tipo de dado Alarme_t , as suas funções de CRUD e suas funções auxiliares	15
Equipamento.h	Arquivo contendo o tipo de dado Equipamento_t , suas funções de CRUD as suas funções auxiliares	18
Interface.h	Arquivo contendo as funções de interface e suas funções auxiliares	21
Sistema.c	Arquivo contendo a função main e todos os includes	28
Sistema.h	Arquivo contendo as definições e tipos de dados utilizados no sistema	29

Capítulo 5

Estruturas

5.1 Referência da Estrutura Alarme_t

Estrutura de dados para o armazenamento e manipulação de alarmes.

```
#include <Alarme.h>
```

Campos de Dados

- char Nome [NOME_MAX]
- int Classificacao
- int N_Serie_Equi
- int identificador
- bool Ativo
- int N_Ativacao
- Data_t DataCadastro
- Data_t DataEntrada
- Data_t DataSaida
- char Descricao [DESC_MAX]

5.1.1 Descrição detalhada

Estrutura de dados para o armazenamento e manipulação de alarmes.

5.1.2 Campos

5.1.2.1 Ativo

```
bool Ativo
```

Boolean que informa se o alarmes está ativo ou não. (TRUE:0 FALSE:1)

5.1.2.2 Classificacao

```
int Classificacao
```

Inteiro contendo a classificação do alarme em 3 possíveis. (ALTA:1 MEDIA:2 BAIXA:3)

5.1.2.3 DataCadastro

```
Data_t DataCadastro
```

Date_t contendo a data em que o alarme foi cadastrado.

5.1.2.4 DataEntrada

```
Data_t DataEntrada
```

Date_t contendo a data em que o alarme foi ativado.

5.1.2.5 DataSaida

```
Data_t DataSaida
```

Date_t contendo a data em que o alarme será desativado.

5.1.2.6 Descricao

```
char Descricao[DESC_MAX]
```

String contendo a descrição do alarme

5.1.2.7 identificador

```
int identificador
```

Inteiro contendo o identificador do alarme.

5.1.2.8 N_Ativacao

```
int N_Ativacao
```

Inteiro contendo o número de vezes que esse alarme já foi ativado.

5.1.2.9 N_Serie_Equi

```
int N_Serie_Equi
```

Inteiro contendo o número de serie do equipamento relacionado ao alarme.

5.1.2.10 Nome

```
char Nome[NOME_MAX]
```

String contendo o nome do alarme.

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

- [Alarme.h](#)

5.2 Referência da Estrutura Data_t

Tipo de dados para datas.

```
#include <Sistema.h>
```

Campos de Dados

- int [Dia](#)
- int [Mes](#)
- int [Ano](#)

5.2.1 Descrição detalhada

Tipo de dados para datas.

Atividades futuras Utilizar a time.h para definir datas.

5.2.2 Campos

5.2.2.1 Ano

```
int Ano
```

5.2.2.2 Dia

```
int Dia
```

5.2.2.3 Mes

```
int Mes
```

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

- [Sistema.h](#)

5.3 Referência da Estrutura Equipamento_t

Estrutura de dados para o armazenamento e manipulação de equipamentos.

```
#include <Equipamento.h>
```

Campos de Dados

- char [Nome](#) [NOME_MAX]
- int [N_Serie](#)
- int [Tipo](#)
- [Data_t](#) [Data](#)
- char [Descricao](#) [DESC_MAX]

5.3.1 Descrição detalhada

Estrutura de dados para o armazenamento e manipulação de equipamentos.

5.3.2 Campos

5.3.2.1 Data

```
Data\_t Data
```

[Data_t](#) contendo a data em que o equipamento foi cadastrado.

5.3.2.2 Descricao

```
char Descricao[DESC\_MAX]
```

String contendo a descrição do equipamento.

5.3.2.3 N_Serie

```
int N_Serie
```

Inteiro contendo o número de serie do equipamento.

5.3.2.4 Nome

```
char Nome[NOME_MAX]
```

String contendo o nome do equipamento.

5.3.2.5 Tipo

```
int Tipo
```

Inteiro contendo o tipo do equipamento. (TENSAO:1 CORRENTE:2 OLEO:3)

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

- [Equipamento.h](#)

5.4 Referência da Estrutura IntAlarm_t

Estrutura de dados que contém um identificador de alarme e um inteiro, utilizada para a ordenação dos alarmes.

```
#include <Sistema.h>
```

Campos de Dados

- int [Id](#)
- int [Val](#)

5.4.1 Descrição detalhada

Estrutura de dados que contém um identificador de alarme e um inteiro, utilizada para a ordenação dos alarmes.

Veja também

[SortInt\(\)](#)

5.4.2 Campos

5.4.2.1 Id

```
int Id
```

Inteiro contendo o identificador do alarme.

5.4.2.2 Val

```
int Val
```

Inteiro contendo o valor a ser comparado.

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

- [Sistema.h](#)

5.5 Referência da Estrutura StrAlarm_t

Estrutura de dados que contém um identificador de alarme e uma string, utilizado para ordenação de strings.

```
#include <Sistema.h>
```

Campos de Dados

- int [Id](#)
- char [Desc](#) [[DESC_MAX](#)]

5.5.1 Descrição detalhada

Estrutura de dados que contém um identificador de alarme e uma string, utilizado para ordenação de strings.

Veja também

[SortString\(\)](#)

5.5.2 Campos

5.5.2.1 Desc

```
char Desc[DESC_MAX]
```

String contendo a descrição do alarme a ser comparado.

5.5.2.2 Id

```
int Id
```

Inteiro contendo o identificador do alarme.

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

- [Sistema.h](#)

Capítulo 6

Arquivos

6.1 Referência do Arquivo Alarme.h

Arquivo contendo o tipo de dado [Alarme_t](#), as suas funções de CRUD e suas funções auxiliares.

Estruturas de Dados

- struct [Alarme_t](#)
Estrutura de dados para o armazenamento e manipulação de alarmes.

Funções

- int [CreateAlarme](#) ([Alarme_t](#) A)
Cria/insere um Alarme object.
- int [ReadAlarme](#) (int Identificador, [Alarme_t](#) *A)
Lê um alarme a partir de um identificador e armazena em A o alarme encontrado.
- int [UpdateAlarme](#) ([Alarme_t](#) A)
Atualiza um alarme.
- int [DeleteAlarme](#) (int Identificador)
Deleta da lista de alarmes um alarme a partir de seu identificador.
- void [String2Alarme_t](#) (char *linha, [Alarme_t](#) *A)
Função que converte uma string em uma estrutura de dados [Alarme_t](#).

6.1.1 Descrição detalhada

Arquivo contendo o tipo de dado [Alarme_t](#), as suas funções de CRUD e suas funções auxiliares.

Autor

Gustavo Correa (gustavocorrea@alunos.utfpr.edu.br)

Versão

0.1

Data

2020-09-27

Copyright

Copyright (c) 2020

6.1.2 Funções

6.1.2.1 CreateAlarme()

```
int CreateAlarme (
    Alarme_t A )
```

Cria/insere um Alarme object.

Parâmetros

A	Recebe um <code>Alarme_t</code> que será inserido a lista de alarmes.
---	---

Retorna

int Retorna TRUE no caso de sucesso e FALSE em caso de falha.

6.1.2.2 DeleteAlarme()

```
int DeleteAlarme (
    int Identificador )
```

Deleta da lista de alarmes um alarme a partir de seu identificador.

Parâmetros

Identificador	Identificador do alarme a ser removido.
---------------	---

Retorna

int Retorna TRUE no caso de sucesso e FALSE em caso de falha.

6.1.2.3 ReadAlarme()

```
int ReadAlarme (
    int Identificador,
    Alarme_t * A )
```

Lê um alarme a partir de um identificador e armazena em A o alarme encontrado.

Parâmetros

<i>Identificador</i>	Identificador que será lido.
<i>A</i>	Ponteiro para um Alarme_t onde ficará armazenado o alarme encontrado.

Retorna

int Retorna TRUE no caso de sucesso e FALSE em caso de falha.

6.1.2.4 String2Alarme_t()

```
void String2Alarme_t (
    char * linha,
    Alarme_t * A )
```

Função que converte uma string em uma estrutura de dados [Alarme_t](#).

Parâmetros

<i>linha</i>	Buffer contendo a string.
<i>A</i>	Estrutura de dados Alarme_t em que será armazenada o conteúdo da string.

Aviso

A string recebida deverá ter os itens separados por tabs ('\t') e finalizados com um '\n' na seguinte ordem: identificador, N_Serie_Equi, Ativo, Nome, N_Ativacao, Classificacao, DataCadastro(Dia, Mes, Ano), Data↵Entrada(Dia, Mes, Ano), DataSaida(Dia, Mes, Ano) e Descricao.

6.1.2.5 UpdateAlarme()

```
int UpdateAlarme (
    Alarme_t A )
```

Atualiza um alarme.

Parâmetros

<i>A</i>	Alarme_t que será atualizado, deve já conter todos os termos que será inseridos.
----------	--

Retorna

int Retorna TRUE no caso de sucesso e FALSE em caso de falha.

6.2 Referência do Arquivo Doc_pages/mainpage.md

6.3 Referência do Arquivo Equipamento.h

Arquivo contendo o tipo de dado [Equipamento_t](#), suas funções de CRUD as suas funções auxiliares.

Estruturas de Dados

- struct [Equipamento_t](#)
Estrutura de dados para o armazenamento e manipulação de equipamentos.

Definições de Tipos

- typedef struct [Equipamento_t](#) [Equipamento_t](#)
Estrutura de dados para o armazenamento e manipulação de equipamentos.

Funções

- int [CreateEquipamento](#) ([Equipamento_t](#) E)
Cria/insere um Equipamento object.
- int [ReadEquipamento](#) (int N_Serie, [Equipamento_t](#) *E)
Lê um Equipamento a partir de um N_Serie e armazena em A o Equipamento encontrado.
- int [UpdateEquipamento](#) ([Equipamento_t](#) E)
Atualiza um Equipamento.
- int [DeleteEquipamento](#) (int N_Serie)
Deleta da lista de Equipamentos um Equipamento a partir de seu N_Serie.
- int [GetInt](#) (int *i, char *linha)
Obtem um inteiro a partir de uma string.
- void [Limpar_buffer](#) (char *buf)
Limpa um buffer.
- void [String2Equipamento_t](#) (char *linha, [Equipamento_t](#) *Equip)
Função que converte uma string em uma estrutura de dados [Equipamento_t](#).

6.3.1 Descrição detalhada

Arquivo contendo o tipo de dado [Equipamento_t](#), suas funções de CRUD as suas funções auxiliares.

Autor

Gustavo Correa (gustavocorrea@alunos.utfpr.edu.br)

Versão

0.1

Data

2020-09-27

Copyright

Copyright (c) 2020

6.3.2 Definições dos tipos

6.3.2.1 Equipamento_t

```
typedef struct Equipamento_t Equipamento_t
```

Estrutura de dados para o armazenamento e manipulação de equipamentos.

6.3.3 Funções

6.3.3.1 CreateEquipamento()

```
int CreateEquipamento (
    Equipamento_t E )
```

Cria/insere um Equipamento object.

Parâmetros

A	Recebe um <code>Equipamento_t</code> que será inserido a lista de Equipamentos.
---	---

Retorna

int Retorna TRUE no caso de sucesso e FALSE em caso de falha.

6.3.3.2 DeleteEquipamento()

```
int DeleteEquipamento (
    int N_Serie )
```

Deleta da lista de Equipamentos um Equipamento a partir de seu N_Serie.

Parâmetros

N_Serie	N_Serie do Equipamento a ser removido.
---------	--

Retorna

int Retorna TRUE no caso de sucesso e FALSE em caso de falha.

6.3.3.3 GetInt()

```
int GetInt (
    int * i,
    char * linha )
```

Obtem um inteiro a partir de uma string.

Parâmetros

<i>i</i>	Contador que aponta a posição da string onde deverá ser lido o inteiro.
<i>linha</i>	Buffer contendo a string onde será lido o inteiro.

Retorna

int Retorna o inteiro lido.

6.3.3.4 Limpar_buffer()

```
void Limpar_buffer (
    char * buf )
```

Limpa um buffer.

Parâmetros

<i>buf</i>	Buffer a ser limpo.
------------	---------------------

6.3.3.5 ReadEquipamento()

```
int ReadEquipamento (
    int N_Serie,
    Equipamento_t * E )
```

Lê um Equipamento a partir de um *N_Serie* e armazena em *A* o Equipamento encontrado.

Parâmetros

<i>N_Serie</i>	<i>N_Serie</i> que será lido.
<i>A</i>	Ponteiro para um Equipamento_t onde ficará armazenado o Equipamento encontrado.

Retorna

int Retorna TRUE no caso de sucesso e FALSE em caso de falha.

6.3.3.6 String2Equipamento_t()

```
void String2Equipamento_t (
    char * linha,
    Equipamento_t * Equip )
```

Função que converte uma string em uma estrutura de dados [Equipamento_t](#).

Parâmetros

<i>linha</i>	Buffer contendo a string
<i>A</i>	Estrutura de dados Equipamento_t em que será armazenada o conteúdo da string.

Aviso

A string recebida deverá ter os itens separados por tabs ('\t') e finalizados com um '\n' na seguinte ordem: N_Serie, Nome, Tipo, Data(Dia, Mes e Ano) e Descrição.

6.3.3.7 UpdateEquipamento()

```
int UpdateEquipamento (
    Equipamento_t E )
```

Atualiza um Equipamento.

Parâmetros

<i>A</i>	Equipamento_t que será atualizado, deve já conter todos os termos que será inseridos.
----------	---

Retorna

int Retorna TRUE no caso de sucesso e FALSE em caso de falha.

6.4 Referência do Arquivo Interface.h

Arquivo contendo as funções de interface e suas funções auxiliares.

Funções

- int [GetSize](#) (char *A)
GetSize Obter tamanho. Função que obtém a quantidade de elementos de um arquivo.
- void [SortString](#) ([StrAlarm_t](#) vetor[], int inicio, int fim)

- Função de ordenação de strings em um vetor.*
- void [SortInt](#) ([IntAlarm_t](#) vetor[], int inicio, int fim)
Função de ordenação de inteiros em um vetor.
- void [ImprimirEquipamento](#) ([Equipamento_t](#) *Eq)
Função que imprime na tela todas as informações de um equipamento.
- void [ImprimirTudoEquipamentos](#) ()
Função que imprime toda a lista de equipamentos.
- void [Atualiza_Equipamento](#) ()
Função que faz a atualização de um equipamento para o usuário.
- void [Criar_Equipamento](#) ()
Função que cria/insere um equipamento na lista para o usuário.
- void [Remover_Equipamento](#) ()
Função que remove um equipamento para o usuário.
- void [Pesquisa_Equipamento](#) ()
Função que pesquisa um equipamento para o usuário.
- void [ImprimirAlarme](#) ([Alarme_t](#) *A)
Função que imprime na tela todas as informações de um Alarme.
- void [ImprimirTudoAlarmes](#) ()
Função que imprime toda a lista de Alarmes.
- void [Atualiza_Alarme](#) ()
Função que atualiza um alarme para o usuário.
- void [Criar_Alarme](#) ()
Função que cria/insere um novo alarme para o usuário.
- void [Remover_Alarme](#) ()
Função que remove um alarme da lista para o usuario.
- void [Pesquisar_Alarme](#) ()
Função que pesquisa um alarme para o usuário.
- void [ImprimirAtuado](#) ([Alarme_t](#) *A)
Função que imprime para o usuário um Alarme que esteja atuado.
- void [ImprimirTodosAtuados](#) ()
Função que imprime todos os alarmes atuados.
- void [Ativar_Alarmes](#) ()
Função que ativa um alarme para o usuário.
- void [Desativar_Alarmes](#) ()
Função que desativa um alarme para o usuário.
- void [OrdenarClassificacao](#) ()
Função que mostra de forma ordenada pela classificação a lista de alarmes.
- void [Ordenar_Desc](#) ()
Função que mostra de forma ordenada pela descrição a lista de alarmes.
- void [OrdenarAtuacao](#) ()
Função que mostra de forma ordenada pela quantidade de vezes que o alarme foi atuado a lista de alarmes.
- void [Filtrar](#) ()
Função que filtra os alarmes e mostra somente aqueles que contem na descrição uma palavra que o usuário passar.
- void [Menu](#) ()
Função que gera o menu.

6.4.1 Descrição detalhada

Arquivo contendo as funções de interface e suas funções auxiliares.

Autor

Gustavo Correa (gustavocorrea@alunos.utfpr.edu.br)

Versão

0.1

Data

2020-09-27

Copyright

Copyright (c) 2020

6.4.2 Funções

6.4.2.1 Ativar_Alarmes()

```
void Ativar_Alarmes ( )
```

Função que ativa um alarme para o usuário.

6.4.2.2 Atualiza_Alarme()

```
void Atualiza_Alarme ( )
```

Função que atualiza um alarme para o usuário.

6.4.2.3 Atualiza_Equipamento()

```
void Atualiza_Equipamento ( )
```

Função que faz a atualização de um equipamento para o usuário.

6.4.2.4 Criar_Alarme()

```
void Criar_Alarme ( )
```

Função que cria/insere um novo alarme para o usuário.

6.4.2.5 Criar_Equipamento()

```
void Criar_Equipamento ( )
```

Função que cria/insere um equipamento na lista para o usuário.

6.4.2.6 Desativar_Alarmes()

```
void Desativar_Alarmes ( )
```

Função que desativa um alarme para o usuário.

6.4.2.7 Filtrar()

```
void Filtrar ( )
```

Função que filtra os alarmes e mostra somente aqueles que contem na descrição uma palavra que o usuário passar.

6.4.2.8 GetSize()

```
int GetSize (
    char * A )
```

GetSize Obter tamanho. Função que obtem a quantidade de elementos de um arquivo.

Parâmetros

A	Caminho para o arquivo que se deseja saber o tamanho.
---	---

Retorna

int Retorna a quantidade de elementos que o arquivo possui.

6.4.2.9 ImprimirAlarme()

```
void ImprimirAlarme (
    Alarme_t * A )
```

Função que imprime na tela todas as informações de um Alarme.

Parâmetros

A	Ponteiro para um Alarme_t que será impresso.
---	--

6.4.2.10 ImprimirAtuado()

```
void ImprimirAtuado (
    Alarme_t * A )
```

Função que imprime para o usuário um Alarme que esteja atuado.

Parâmetros

A	Ponteiro para um Alarme_t que será impresso.
---	--

6.4.2.11 ImprimirEquipamento()

```
void ImprimirEquipamento (
    Equipamento_t * Eq )
```

Função que imprime na tela todas as informações de um equipamento.

Parâmetros

Eq	Ponteiro para um Equipamento_t que será impresso.
----	---

6.4.2.12 ImprimirTodosAtuados()

```
void ImprimirTodosAtuados ( )
```

Função que imprime todos os alarmes atuados.

6.4.2.13 ImprimirTudoAlarmes()

```
void ImprimirTudoAlarmes ( )
```

Função que imprime toda a lista de Alarmes.

6.4.2.14 ImprimirTudoEquipamentos()

```
void ImprimirTudoEquipamentos ( )
```

Função que imprime toda a lista de equipamentos.

6.4.2.15 Menu()

```
void Menu ( )
```

Função que gera o menu.

6.4.2.16 Ordenar_Desc()

```
void Ordenar_Desc ( )
```

Função que mostra de forma ordenada pela descrição a lista de alarmes.

6.4.2.17 OrdenarAtuacao()

```
void OrdenarAtuacao ( )
```

Função que mostra de forma ordenada pela quantidade de vezes que o alarme foi atuado a lista de alarmes.

6.4.2.18 OrdenarClassificacao()

```
void OrdenarClassificacao ( )
```

Função que mostra de forma ordenada pela classificação a lista de alarmes.

6.4.2.19 Pesquisa_Equipamento()

```
void Pesquisa_Equipamento ( )
```

Função que pesquisa um equipamento para o usuário.

6.4.2.20 Pesquisar_Alarme()

```
void Pesquisar_Alarme ( )
```

Função que pesquisa um alarme para o usuário.

6.4.2.21 Remover_Alarme()

```
void Remover_Alarme ( )
```

Função que remove um alarme da lista para o usuário.

6.4.2.22 Remover_Equipamento()

```
void Remover_Equipamento ( )
```

Função que remove um equipamento para o usuário.

6.4.2.23 SortInt()

```
void SortInt (
    IntAlarm_t vetor[],
    int inicio,
    int fim )
```

Função de ordenação de inteiros em um vetor.

Parâmetros

<i>vetor</i>	Vetor a ser ordenado.
<i>inicio</i>	Início do vetor.
<i>fim</i>	Fim do vetor.

6.4.2.24 SortString()

```
void SortString (
    StrAlarm_t vetor[],
    int inicio,
    int fim )
```

Função de ordenação de strings em um vetor.

Parâmetros

<i>vetor</i>	Vetor a ser ordenado.
<i>inicio</i>	Início do vetor.
<i>fim</i>	Fim do vetor.

6.5 Referência do Arquivo Sistema.c

Arquivo contendo a função main e todos os includes.

```
#include <stdio.h>
#include <stdlib.h>
#include <stddef.h>
#include <time.h>
#include <string.h>
#include <errno.h>
#include "Sistema.h"
#include "Equipamento.h"
#include "Alarme.h"
#include "Interface.h"
```

Funções

- int `main` ()

6.5.1 Descrição detalhada

Arquivo contendo a função main e todos os includes.

Autor

Gustavo Correa (gustavocorrea@alunos.utfpr.edu.br)

Versão

0.1

Data

2020-09-26

Copyright

Copyright (c) 2020

6.5.2 Funções

6.5.2.1 main()

```
int main ( )
```

6.6 Referência do Arquivo Sistema.h

Arquivo contendo as definições e tipos de dados utilizados no sistema.

Estruturas de Dados

- struct [StrAlarm_t](#)
Estrutura de dados que contém um identificador de alarme e uma string, utilizado para ordenação de strings.
- struct [IntAlarm_t](#)
Estrutura de dados que contém um identificador de alarme e um inteiro, utilizada para a ordenação dos alarmes.
- struct [Data_t](#)
Tipo de dados para datas.

Definições e Macros

- #define [BYTES_MAX](#) 512
Define o tamanho maximo dos buffers.
- #define [INT_MAX](#) 10
Define o tamanho do buffer que armazena inteiros.
- #define [NOME_MAX](#) 128
Define o tamanho da string de nome.
- #define [DESC_MAX](#) 256
Define o tamanho da string de descrição.
- #define [TRUE](#) 0
- #define [FALSE](#) 1
- #define [EQUIP_FILE](#) "Equipamentos.tsv"
Define o caminho e o nome do arquivo que contém a lista de equipamentos cadastrados no sistema.
- #define [ALARME_FILE](#) "Alarmes.tsv"
Define o caminho e o nome do arquivo que contém a lista de alarmes cadastrados no sistema.
- #define [TEMP](#) "Temporario.temp"
Define o caminho e o nome do arquivo temporário que é utilizado no sistema.

Definições de Tipos

- typedef int [bool](#)
- typedef struct [StrAlarm_t](#) [StrAlarm_t](#)
Estrutura de dados que contém um identificador de alarme e uma string, utilizado para ordenação de strings.
- typedef struct [IntAlarm_t](#) [IntAlarm_t](#)
Estrutura de dados que contém um identificador de alarme e um inteiro, utilizada para a ordenação dos alarmes.
- typedef struct [Data_t](#) [Data_t](#)
Tipo de dados para datas.

Enumerações

- enum { ALTA = 1, MEDIA, BAIXA }
Enum contendo a definição das classificações dos alarmes.
- enum { TENSAO = 1, CORRENTE, OLEO }
Enum contendo a definição dos tipos de equipamentos.

6.6.1 Descrição detalhada

Arquivo contendo as definições e tipos de dados utilizados no sistema.

Autor

Gustavo Correa (gustavocorrea@alunos.utfpr.edu.br)

Versão

0.1

Data

2020-09-26

Copyright

Copyright (c) 2020

6.6.2 Definições e macros

6.6.2.1 ALARME_FILE

```
#define ALARME_FILE "Alarmes.tsv"
```

Define o caminho e o nome do arquivo que contém a lista de alarmes cadastrados no sistema.

6.6.2.2 BYTES_MAX

```
#define BYTES_MAX 512
```

Define o tamanho maximo dos buffers.

6.6.2.3 DESC_MAX

```
#define DESC_MAX 256
```

Define o tamanho da string de descrição.

6.6.2.4 EQUIP_FILE

```
#define EQUIP_FILE "Equipamentos.tsv"
```

Define o caminho e o nome do arquivo que contém a lista de equipamentos cadastrados no sistema.

6.6.2.5 FALSE

```
#define FALSE 1
```

6.6.2.6 INT_MAX

```
#define INT_MAX 10
```

Define o tamanho do buffer que armazena inteiros.

6.6.2.7 NOME_MAX

```
#define NOME_MAX 128
```

Define o tamanho da string de nome.

6.6.2.8 TEMP

```
#define TEMP "Temporario.temp"
```

Define o caminho e o nome do arquivo temporário que é utilizado no sistema.

6.6.2.9 TRUE

```
#define TRUE 0
```

6.6.3 Definições dos tipos

6.6.3.1 bool

```
typedef int bool
```

6.6.3.2 Data_t

```
typedef struct Data_t Data_t
```

Tipo de dados para datas.

Atividades futuras Utilizar a time.h para definir datas.

6.6.3.3 IntAlarm_t

```
typedef struct IntAlarm_t IntAlarm_t
```

Estrutura de dados que contém um identificador de alarme e um inteiro, utilizada para a ordenação dos alarmes.

Veja também

[SortInt\(\)](#)

6.6.3.4 StrAlarm_t

```
typedef struct StrAlarm_t StrAlarm_t
```

Estrutura de dados que contém um identificador de alarme e uma string, utilizado para ordenação de strings.

Veja também

[SortString\(\)](#)

6.6.4 Enumerações

6.6.4.1 anonymous enum

```
anonymous enum
```

Enum contendo a definição das classificações dos alarmes.

Enumeradores

ALTA	Classificação alta para o alarme.
MEDIA	Classificação média para o alarme.
BAIXA	Classificação baixa para o alarme.

6.6.4.2 anonymous enum

`anonymous enum`

Enum contendo a definição dos tipos de equipamentos.

Enumeradores

TENSAO	Equipamento do tipo tensão.
CORRENTE	Equipamento do tipo corrente.
OLEO	Equipamento do tipo óleo.

Índice Remissivo

Alarme.h, [15](#)
 CreateAlarme, [16](#)
 DeleteAlarme, [16](#)
 ReadAlarme, [16](#)
 String2Alarme_t, [17](#)
 UpdateAlarme, [17](#)
ALARME_FILE
 Sistema.h, [30](#)
Alarme_t, [9](#)
 Ativo, [9](#)
 Classificacao, [9](#)
 DataCadastro, [10](#)
 DataEntrada, [10](#)
 DataSaida, [10](#)
 Descricao, [10](#)
 identificador, [10](#)
 N_Ativacao, [10](#)
 N_Serie_Equi, [10](#)
 Nome, [10](#)
ALTA
 Sistema.h, [33](#)
Ano
 Data_t, [11](#)
Ativar_Alarmes
 Interface.h, [23](#)
Ativo
 Alarme_t, [9](#)
Atualiza_Alarme
 Interface.h, [23](#)
Atualiza_Equipamento
 Interface.h, [23](#)

BAIXA
 Sistema.h, [33](#)
bool
 Sistema.h, [32](#)
BYTES_MAX
 Sistema.h, [30](#)

Classificacao
 Alarme_t, [9](#)
CORRENTE
 Sistema.h, [33](#)
CreateAlarme
 Alarme.h, [16](#)
CreateEquipamento
 Equipamento.h, [19](#)
Criar_Alarme
 Interface.h, [23](#)
Criar_Equipamento

Interface.h, [24](#)
Data
 Equipamento_t, [12](#)
Data_t, [11](#)
 Ano, [11](#)
 Dia, [11](#)
 Mes, [11](#)
 Sistema.h, [32](#)
DataCadastro
 Alarme_t, [10](#)
DataEntrada
 Alarme_t, [10](#)
DataSaida
 Alarme_t, [10](#)
DeleteAlarme
 Alarme.h, [16](#)
DeleteEquipamento
 Equipamento.h, [19](#)
Desativar_Alarmes
 Interface.h, [24](#)
Desc
 StrAlarm_t, [14](#)
DESC_MAX
 Sistema.h, [30](#)
Descricao
 Alarme_t, [10](#)
 Equipamento_t, [12](#)
Dia
 Data_t, [11](#)
Doc_pages/mainpage.md, [18](#)

EQUIP_FILE
 Sistema.h, [31](#)
Equipamento.h, [18](#)
 CreateEquipamento, [19](#)
 DeleteEquipamento, [19](#)
 Equipamento_t, [19](#)
 GetInt, [19](#)
 Limpar_buffer, [20](#)
 ReadEquipamento, [20](#)
 String2Equipamento_t, [21](#)
 UpdateEquipamento, [21](#)
Equipamento_t, [12](#)
 Data, [12](#)
 Descricao, [12](#)
 Equipamento.h, [19](#)
 N_Serie, [12](#)
 Nome, [13](#)
 Tipo, [13](#)

FALSE
 Sistema.h, 31
 Filtrar
 Interface.h, 24

 GetInt
 Equipamento.h, 19
 GetSize
 Interface.h, 24

 Id
 IntAlarm_t, 13
 StrAlarm_t, 14
 identificador
 Alarme_t, 10
 ImprimirAlarme
 Interface.h, 24
 ImprimirAtuado
 Interface.h, 25
 ImprimirEquipamento
 Interface.h, 25
 ImprimirTodosAtuados
 Interface.h, 25
 ImprimirTudoAlarmes
 Interface.h, 25
 ImprimirTudoEquipamentos
 Interface.h, 26
 INT_MAX
 Sistema.h, 31
 IntAlarm_t, 13
 Id, 13
 Sistema.h, 32
 Val, 14
 Interface.h, 21
 Ativar_Alarmes, 23
 Atualiza_Alarme, 23
 Atualiza_Equipamento, 23
 Criar_Alarme, 23
 Criar_Equipamento, 24
 Desativar_Alarmes, 24
 Filtrar, 24
 GetSize, 24
 ImprimirAlarme, 24
 ImprimirAtuado, 25
 ImprimirEquipamento, 25
 ImprimirTodosAtuados, 25
 ImprimirTudoAlarmes, 25
 ImprimirTudoEquipamentos, 26
 Menu, 26
 Ordenar_Desc, 26
 OrdenarAtuacao, 26
 OrdenarClassificacao, 26
 Pesquisa_Equipamento, 26
 Pesquisar_Alarme, 27
 Remover_Alarme, 27
 Remover_Equipamento, 27
 SortInt, 27
 SortString, 27

 Limpar_buffer
 Equipamento.h, 20

 main
 Sistema.c, 29
 MEDIA
 Sistema.h, 33
 Menu
 Interface.h, 26
 Mes
 Data_t, 11

 N_Ativacao
 Alarme_t, 10
 N_Serie
 Equipamento_t, 12
 N_Serie_Equi
 Alarme_t, 10
 Nome
 Alarme_t, 10
 Equipamento_t, 13
 NOME_MAX
 Sistema.h, 31

 OLEO
 Sistema.h, 33
 Ordenar_Desc
 Interface.h, 26
 OrdenarAtuacao
 Interface.h, 26
 OrdenarClassificacao
 Interface.h, 26

 Pesquisa_Equipamento
 Interface.h, 26
 Pesquisar_Alarme
 Interface.h, 27

 ReadAlarme
 Alarme.h, 16
 ReadEquipamento
 Equipamento.h, 20
 Remover_Alarme
 Interface.h, 27
 Remover_Equipamento
 Interface.h, 27

 Sistema.c, 28
 main, 29
 Sistema.h, 29
 ALARME_FILE, 30
 ALTA, 33
 BAIXA, 33
 bool, 32
 BYTES_MAX, 30
 CORRENTE, 33
 Data_t, 32
 DESC_MAX, 30
 EQUIP_FILE, 31
 FALSE, 31

- INT_MAX, [31](#)
- IntAlarm_t, [32](#)
- MEDIA, [33](#)
- NOME_MAX, [31](#)
- OLEO, [33](#)
- StrAlarm_t, [32](#)
- TEMP, [31](#)
- TENSAO, [33](#)
- TRUE, [31](#)
- SortInt
 - Interface.h, [27](#)
- SortString
 - Interface.h, [27](#)
- StrAlarm_t, [14](#)
 - Desc, [14](#)
 - Id, [14](#)
 - Sistema.h, [32](#)
- String2Alarme_t
 - Alarme.h, [17](#)
- String2Equipamento_t
 - Equipamento.h, [21](#)
- TEMP
 - Sistema.h, [31](#)
- TENSAO
 - Sistema.h, [33](#)
- Tipo
 - Equipamento_t, [13](#)
- TRUE
 - Sistema.h, [31](#)
- UpdateAlarme
 - Alarme.h, [17](#)
- UpdateEquipamento
 - Equipamento.h, [21](#)
- Val
 - IntAlarm_t, [14](#)