Ano Letivo: 2023/2024

Relatório Final

POO

Trabalho realizado por:

Martim Alexandre Vieira Antunes

nº: 2022141890 **Curso:** LEI

Pedro Lino Neves Faneca

nº: 2022134142 **Curso:** LEI

Índice

1.Introdução

O projeto em análise é uma implementação de um sistema de automação residencial, onde diversas entidades, como Zonas, Sensores, Regras e Processador, interagem para criar um ambiente automatizado. O código é organizado em várias classes que representam entidades específicas e funcionalidades relacionadas à automação.

2.Classes Necessárias

- **Classe Interação**

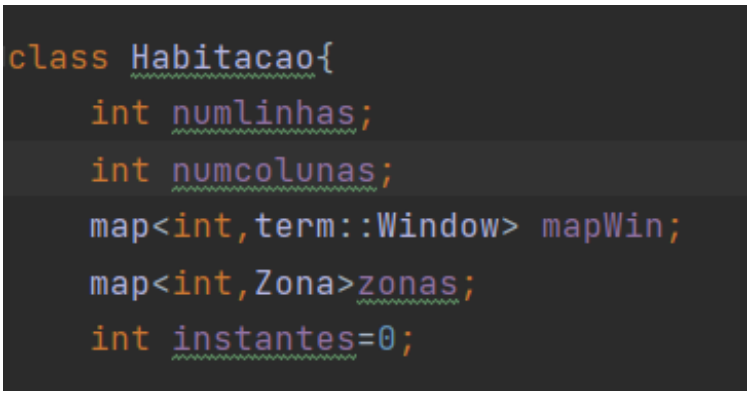
  A classe Interação é responsável por ler os comandos e pela interface com o utilizador.

```
class Interacao{
    Habitacao habitacao;
    Window cmd;
    Window infos;
    static const std::map<std::string, int> comandos;

public:
    Interacao();
     void leComando ();
    void checkcomando(string frase);
    bool leFicheiro(const string &ficheiro,Window &infos);
```

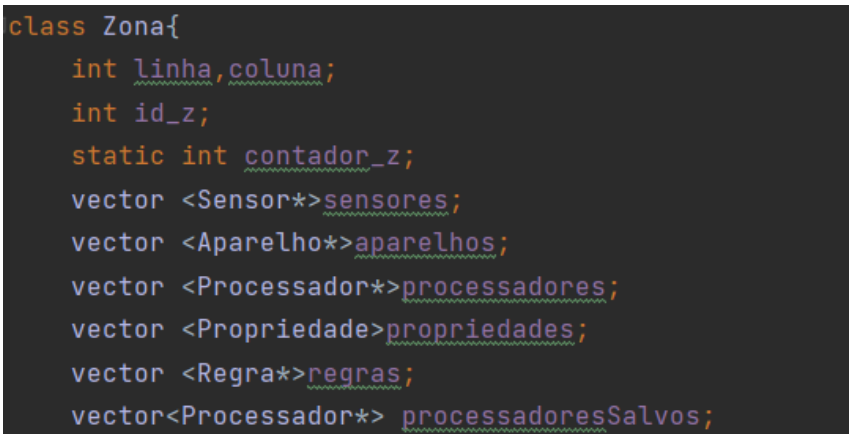
- **Classe Habitacão**

A classe Habitação é a classe principal. Esta classe permite criar uma habitação, criar zonas, procurar por uma zona específica e adicionar sensores, aparelhos, processadores em cada zona. Tem o número de linhas e colunas da habitação, um map de janelas e um de zonas.



- **Classe Zona**

  Esta classe permite adicionar os componentes, eliminá-los, listá-los em cada zona, associar as regras aos processadores, adicionar regras de 1 ou 2 parâmetros adicionais, etc. Tem os vetores associados aos sensores,aparelhos,propriedades.



Classe Propriedade

A classe Propriedade tem como herança as diferentes propriedades existentes nas zonas, sendo estas:

- Temperatura

- Fumo
- Humidade
- Luz
- Radiação
- Som
- Vibração

```
class Propriedade{
    string tipo;
    string unidade;
    double valor;
    double valor_min;
    double valor_max;

public:
    Propriedade(string nome, double val, double val_min, double val_max,string unidade);
    Propriedade(const Propriedade &outro);
    string gettipo()const;
    string getunidade()const;
    double getvalor()const;
    bool setvalor(double novovalor);
    void subvalor(double valorSubtrair);
    void addvalor(double valorAdicionar);
    string getString()const;
};
```

- **Classe Sensor**

Esta classe recebe um ponteiro para a propriedade. Serve para verificar qual o valor em cada zona da sua respetiva propriedade.

```
class Sensor{
    string tipo="s";
    int id;
    static int contador;
    Propriedade *propriedade;
public:
    //Sensor(string zona,string chave);
    Sensor(Propriedade *propriedade) : propriedade(propriedade)  {id=++contador;}
    ~Sensor(){delete propriedade;}
    string letra()const;
    int getId()const;
    string getString()const;
    void realizarLeitura();
    double getvalor_prop();
    double setvalor_prop(double novo_valor);
    string getunidade();
    string gettipo()const;
```

- **Classe Aparelho**

Esta classe tem como herança os diferentes aparelhos, que reagem a um comando que altera os valores das propriedades da zona onde estão inseridos, entre eles:

- Aquecedor
- Aspersor
- Lampada
- Refrigerador

```
class Aparelho{
    string nome;
    string tipo;
    string letraMaiuscula="A";
    string letra="a";
    string comando;
    string efeito;
    int id;
    static int contador;
    bool ligado;
    int instante_ligado;
    vector<Propriedade*>propriedades;
```

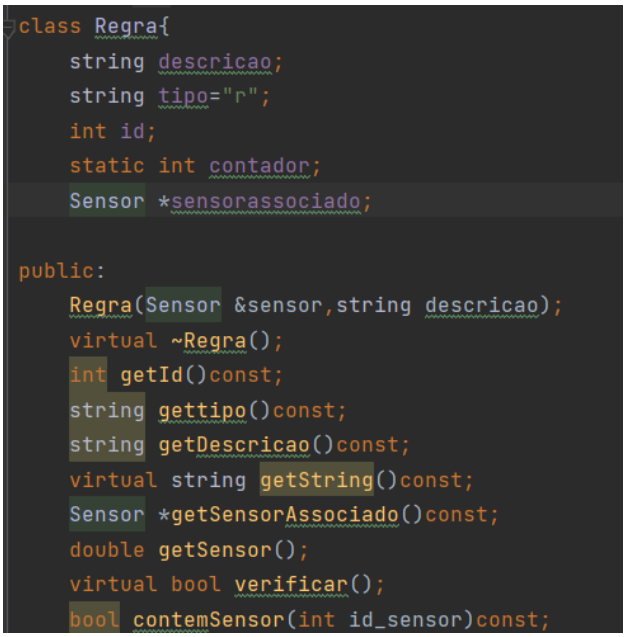
- **Classe Processador**

  A classe Processador é onde é armazenado as regras que ao serem avaliadas transmitem ao aparelho associado, um comando.

```
class Processador{
    string letra="p";
    int id;
    static int contador;
    vector <Regra*> regras;
    Aparelho *aparelho;
    string comandoProc;
    string nome;
```

- **Classe Regra**

Esta classe tem como herança as diferentes regras em que estas verificam o valor lido nos sensores, entre elas:



- Entre
- Fora
- Igual
- Maior
- Menor

3.Divisão da Janela

A interface do usuário é dividida em três partes principais:

1. **Janela de Zonas:**
    a. Exibe informações sobre as zonas habitáveis e seus componentes.
2. **Janela de Informações:**
    a. Fornece informações detalhadas sobre a zona ou componente selecionado.
3. **Janela de Comandos:**
    a. Aceita comandos do utilizador para interação com o sistema.

Essa divisão facilita a visualização e interação com diferentes partes do sistema de automação residencial.

4.Funcionalidades implementadas

Componente do Trabalho	Implementado	Implementado Parcialmente	Não implementado
Interface com o utilizador		✓	
Comando com o utilizador	✓		
Habitação e Zonas	✓		
Propriedades, Aparelhos Processadores e Sensores		✓	
Salvar/Guardar Processadores		✓	
Interações entre os componentes	✓		

O que não conseguimos implementar foi:

- A parte da interação com o utilizador, não conseguimos eliminar da janela quando o componente era eliminado e quando o aparelho ligava.
- Na parte dos aparelhos alterarem as propriedades da zona consoante o número de instantes.
- A parte de salvar as cópias dos processadores;

5.Conclusão

O código é estruturado de maneira organizada, com classes bem definidas para representar as entidades do sistema de automação. As principais funções e a divisão da interface do usuário foram identificadas para fornecer uma visão geral do funcionamento do sistema.

Este relatório é uma introdução ao código existente e vai ser expandido com mais detalhes sobre cada classe, função ou área específica do projeto, conforme necessário.