

Programação Orientada a Objetos

RELATÓRIO DO TRABALHO PRÁTICO

Meta 2



Guilherme de Sousa Camacho – 2021138502

Maria Tavares Reis – 2021144186

2023/2024

Índice

Introdução	3
Estrutura do trabalho	4
Class	4
File.....	5
Lib	5
Estrutura das classes	6
Funcionamento das Classes	7
Simulador	7
Habitação.....	7
Zona	7
Propriedade	8
Componente	8
Processador	9
Regra	9
Sensor.....	9
Aparelho.....	10

Introdução

Neste trabalho pretende construir-se em C++ um simulador de uma habitação controlada por componentes de domótica interligados entre si.

O simulador inclui zonas da habitação, que têm propriedades tais como temperatura, luz, etc., que são inspecionadas por sensores, os quais fornecem os valores lidos a regras que são geridas por processadores de regras que determinam o que fazer em função das leituras dos sensores.

Estrutura do trabalho

O trabalho está organizado em várias pastas para uma melhor organização do mesmo.

Temos duas pastas principais:

- Class
- File
- Lib

Class

A pasta “class” contém várias pastas também, que são as seguintes e estão organizadas da seguinte forma:

- Simulador
- Habitação
- Zona
- Propriedade
 - Fumo
 - Humidade
 - Luz
 - Radiação
 - Som
 - Temperatura
 - Vibração
- Componente
 - Aparelho
 - Aquecedor
 - Aspersor
 - Lampada
 - Refrigerador
 - Processador

- Regra
- Sensor
 - SensorFumo
 - SensorHumidade
 - SensorLuminosidade
 - SensorMovimento
 - SensorRadiação
 - SensorSom
 - SensorTemperatura

File

A pasta “File” contém os ficheiros de texto onde o comando “exec” vai buscar para facilitar a introdução de comando.

Lib

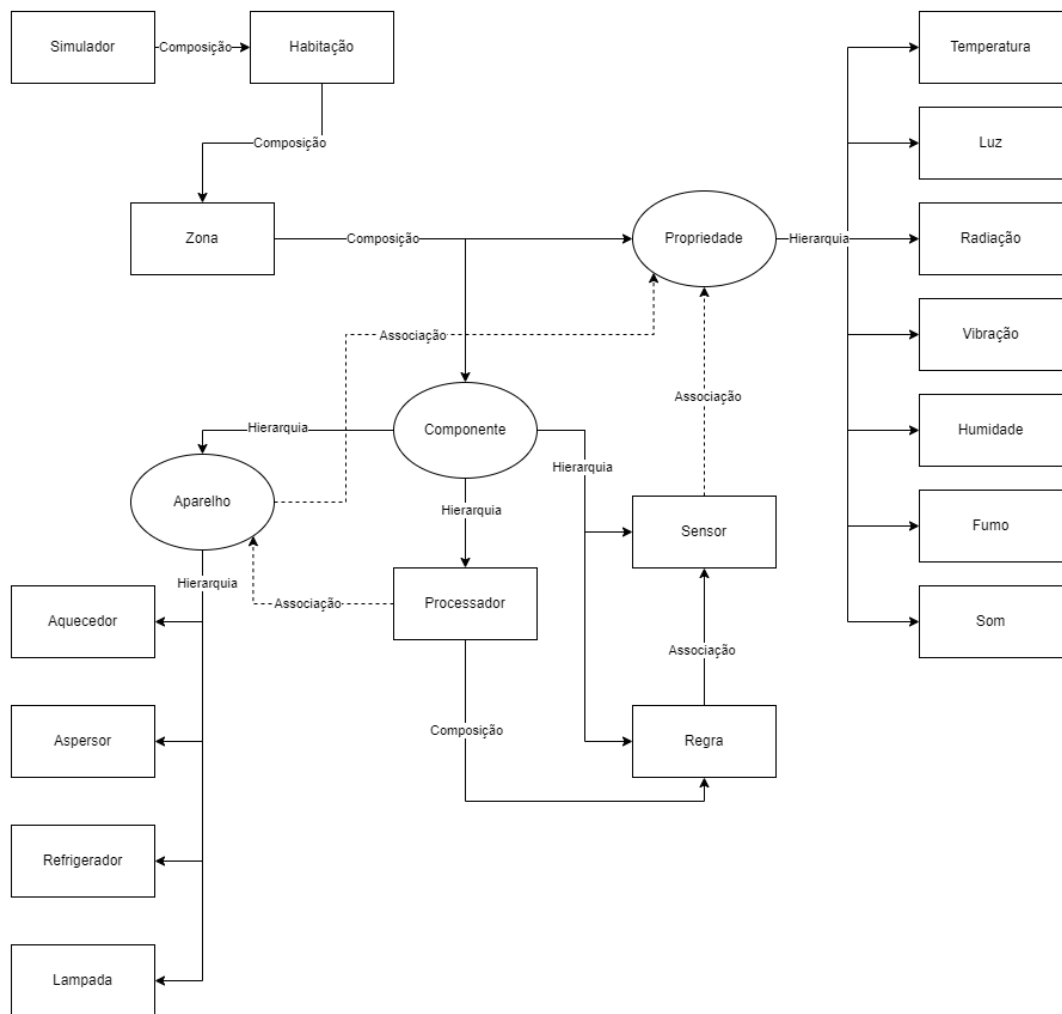
A pasta “lib” contém duas pastas que estão organizadas da seguinte forma:

- Terminal
 - curses.h
 - Terminal.cpp
 - Terminal.h
- Utils
 - utils.cpp
 - utils.h

Estrutura das classes

O trabalho segue com a estrutura de classe do diagrama abaixo.

Os retângulos representam classes e as elipses as classes abstratas, por outras palavras, classes para criação de objetos e classes base para outras classes, respetivamente.



Funcionamento das Classes

Simulador

Class principal do trabalho onde armazena a Habitação e a interface com o utilizador (*UI*).

A Habitação é um objeto dinâmico, ou seja, só é construída em *runtime* quando é introduzido o comando para criar uma Habitação. A *UI* está desenvolvida para mudar de “resolução”, por outras palavras, é possível personalizar o tamanho das janelas para o terminal antes de compilado.

Habitação

Class onde fica armazenada a matriz das Zonas, o tamanho da matriz (altura e largura) e o contador para o *ID* das Zonas.

A matriz é de ponteiros de Zonas para ter Zonas como objetos dinâmicos, por outras palavras, só é criada a Zona quando é pedido.

Zona

Class onde ficam a lista dos componentes e das Propriedades de ambiente.

Cada Zona tem o seu *ID* único e os objetos das suas listas só existem dentro da Zona que foi chamado.

Propriedade

Class abstrata para a criação de todas as classes de Propriedade de ambiente na tabela do enunciado e as suas respectivas restrições.

Funciona como chave-valor e contem uma avaliação de restrição consoante a configuração que o construtor do objeto está feito.

As Propriedades são criadas quando uma Zona é criada, pois uma Zona tem de ter as propriedades todas que a tabela do enunciado mostrou.

Componente

Class abstrata para a criação das classes de Componente que serão os Aparelhos (class abstrata também), Sensores, Processadores de regras e Regras.

Os Componentes contem um *ID* diferente do normal, onde é a junção do tipo de componente com um identificador único a nível de Componentes.

Para uma melhor visualização optamos por dividir o *ID* em duas partes para ajudar na criação dos objetos dos componentes e uma melhor validação.

O *ID* será usada como forma de *output* das janelas das zonas e *input* dos comandos.

Processador

Class onde tem uma lista de Regras para a sua configuração e uma lista de Aparelho que vai comunicar quando todas as regras são ativadas.

As regras só existem quando existe um processador, logo as regras são um conjunto de uma composição da class Regra.

Os aparelhos são criados fora processador, assim o Processador fica com uma relação de associação com o Aparelho.

Regra

Class onde efetua leitura ao sensor que está “ligado” a cada iteração e valida a condição interna que tem.

A Regras tem de estar associada com um Processador e com um Sensor, logo este será um objeto criado um último lugar assim pode garantir que os objetos Sensores e Processadores existem podendo assim usando as variáveis de referência com maior facilidade de recolher de valores.

Sensor

Class onde efetua leitura à propriedade de ambiente associada a ela.

Como as Propriedades existem quando uma Zona é criada, sabemos que a relação entre Sensor e Propriedade será de associação e com o uso de referência pelo mesmo motivo que a relação da Regra e Sensor.

Aparelho

Class abstrata para a criação das class Aquecedor, Aspersor Refrigerador e Lampada.

A class tem o propósito de modificar os valores da propriedade associada a ele e como explicado anteriormente, será uma relação de associação.