

Universidade do Minho

UNIVERSIDADE DO MINHO

LICENCIATURA EM CIÊNCIAS DA COMPUTAÇÃO

Relatório do projeto: SmartDevices, SmartHouses e Controlo/Eficiência Energética

POO- Trabalho Prático  
Grupo 10

Maio de 2022

Uma imagem com pessoa, parede, homem, interior

Descrição gerada automaticamenteUma imagem com pessoa, parede, interior, propriedade

Descrição gerada automaticamenteUma imagem com pessoa, parede, vestuário

Descrição gerada automaticamente

Catarina Quintas João Guedes Pedro Martins

A91650 A94013 A91681

Índice

1. **Introdução** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .

1. **Classes** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
   1. SmartDevice . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
   2. SmartBulb . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
   3. SmartCamera . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
   4. SmartSpeaker . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
   5. Casa. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
   6. FornecedorEnergia . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
   7. Ambient . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
   8. Invoice. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
   9. Parser. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .

2.10. Main. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .

1. **Exceptions** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
   1. DeviceExistsInDivisionExceptions . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
   2. DivisionExistsExceptions. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
   3. HouseNotFoundExceptions . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
2. **Estrutura do projeto** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
3. **Diagrama de classes** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
4. **Conclusão**. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .

**Capítulo 1**

**Introdução**

**No final escrevo**

**Capítulo 2**

**Classes**

**2.1 SmartDevice**

public enum Status{

OFF,

ON

}

private String factoryID; //Código único definido pelo fabricante   
private double mCost; // Custo de instalação   
private Status status; // Status: OFF/ ON do smartDevice   
protected long start; // Ligar o smartDevice  
protected long finish; // Desligar o smartDevice  
protected long timeElapsed; // Tempo percorrido com o SmartDevice ligado

Começamos por fazer uma classe mais geral, onde temos os identificadores gerais representados a cima. Com isto podemos proceder à criação de um dispositivo mais genérico onde identificamos o seu FactoryId, mCost, Status entre outros.

**2.2 SmartBulb**

public enum LightMode{

NEUTRAL, ( 10 w )

WARM, ( 15 w )

COLD ( 20 w )   
}

private LightMode mode; // Tonalidade da luz

private int dimension; // Dimensão

private double dailyConsumption; // Consumo diário

”SmartBulb” é uma subclasse de ”SmartDevice”, que cria os SmartDevices do tipo SmartBulb. Nesta subclasse, dependendo da tonalidade (Neutral/Warm/Cold ) da lâmpada, iremos obter diferentes consumos.   
Por exemplo: Considerando que a lâmpada está no modo “Neutral” ( 10w) , o consumo da energia irá ser dado pela fórmula:

* aux = (double)( 15 \* (dimension\* dimension))/1000;
  1. **SmartCamera**

private int resolution; //resolução da camera

private double fileSize; // tamanho do ficheiro

protected double dailyConsumption; // consumo diário da camera

“SmartCamera” é a subclasse de SmartDevice, onde são criados os SmartDevices do tipo SmartCamera, adicionando às variáveis de instância da sua superclasse: a variável resolução da camera, tamanho do ficheiro e o consumo diário da camera

* 1. **SmartSpeaker**

private int volume;

private String radio;

private String brand;

private double dailyConsumption;

“SmartSpeaker” é a subclasse de SmartDevice, onde são criados os dispositivos do tipo SmartSpeaker. As variáveis de instância desta classe são: volume, radio, marca e o consumo diário.

* 1. **Casa**

private String owner;  
private int NIF;  
private HashMap<String,ArrayList<SmartDevice>>divisions;  
private String provider;

Nesta classe são guardados todas as informações de uma casa, nomeadamente o nome e o nif único de cada proprietário, os Smartdevices de cada divisão e por último o fornecedor. Para cada divisão, é listada uma sequência dos dispositivos que estão registados nessa mesma divisão.

* 1. **FornecedorEnergia**

private String company;  
private double dailyEnergyCost;  
private double tax;

“FornecedorEnergia” é uma classe que providencia energia para as casas, sendo que cada casa pode escolher o seu próprio fornecedor. Nesta classe existe 2 variaveis de instância do tipo double que correspondem ao iva e o custo de Kwh por dia.

* 1. **Ambient**

private Calendar calendar;

“Ambient” é a classe responsável pela gestão do tempo, ou seja, a partir desta classe iremos poder avançar no tempo para programar o ON/OFF dos dispositivos.

**2.8 Invoice**

private String code;  
private Date date;  
private double consumoTotal;  
private FornecedorEnergia Provider;  
private Casa house;  
private ArrayList <String> codeIDs;

“Invoice” é a classe onde irão ser criadas as faturas. Esta classe irá ter toda a informação de uma fatura de energia relativa a uma casa num determinado período de tempo, tendo como informação o código, data, consumo total de energia, fornecedor, casa e o codeIDS

**2.9 Parser**

”Parser” é uma classe feita pela equipa de docente da disciplina, mas tendo algumas alterações da nossa parte para adaptar ao nosso código. Esta classe vai permitir importar os dados de um ficheiro para o programa estando preparada para qualquer eventualidade de erro no ficheiro a ler, tornando assim o ambiente do programa mais seguro e dinâmico.

**2.10 Main**

**Capítulo 3**

**Exceptions**

**3.1 DeviceExistsInDivisionExceptions**

Esta exception foi criada no âmbito de quando o dispositivo já está definido e já existe na divisão, alertando o utilizador para este problema.

**3.2 DivisionExistsExceptions**

Esta exception foi criada no âmbito de quando a divisão já está definida e impede a sobreposição de comandos sobre a mesma, alertando o utilizador para este problema.

**3.3 HouseNotFoundExceptions**

Esta exception foi criada no âmbito de quando a casa não está definida ou quando é definida incorretamente, alertando o utilizador para este problema.

**Capítulo 4**

**Estrutura do projeto**

O nosso projeto segue a estrutura Model View Controller (MVC), estando por isso organizado em três camadas:

* A camada de dados (o modelo) é composta pelas Classes SmartDevice, SmartSpeaker, SmartBulb, SmartCamera, Casa e ForcenedorEnergia.
* A camada de interação com o utilizador (a vista, ou apresentação) é composta unicamente pela classe View.
* A camada de controlo do fluxo do programa (o controlador) é composta pela classe Controller.

Todo o projeto baseia-se na ideia de encapsulamento.

**Capítulo 5**

**Diagrama de Classes**

Diagrama de classes do programa, gerado pelo IntelliJ

**Menu**

Esta classe cria o menu inicial da aplicação.

**Capítulo 6**

**Conclusão**

Em suma, consideramos que a realização deste projeto foi bastante vantajosa, uma vez que não só nos possibilitou uma melhor consolidação dos tópicos abordados nas aulas, como também permitiu-nos aplicar esses conhecimentos numa aplicação em java de maior escala.

Consideramos, ainda, que conseguimos cumprir os objetivos propostos, no entanto, existem melhorias possíveis a realizar, tais como dfjkjjjbkbkvyucuiu