

# Projecto prático de Programação Orientada aos Objectos

LEI + LCC

edição 21/22

## Conteúdo

<b>1</b>	<b>Introdução</b>	<b>1</b>
1.1	Os <i>SmartDevice</i> . . . . .	1
1.2	As casas . . . . .	2
1.3	Os comercializadores de energia . . . . .	2
<b>2</b>	<b>O Funcionamento do programa</b>	<b>2</b>
<b>3</b>	<b>Requisitos</b>	<b>3</b>
3.1	Requisitos base de gestão das entidades . . . . .	3
3.2	Efectuar estatísticas sobre o estado do programa . . . . .	3
3.3	Alterar os operadores e os dispositivos de uma casa . . . . .	4
3.4	Automatizar a simulação . . . . .	4
<b>4</b>	<b>Relatório</b>	<b>4</b>
<b>5</b>	<b>Salvaguarda do estado da aplicação</b>	<b>4</b>
<b>6</b>	<b>Cronograma</b>	<b>5</b>

## 1 Introdução

Considere que se pretende construir um sistema que monitorize e registre a informação sobre o consumo energético das habitações de uma comunidade. Em cada casa existem um conjunto muito alargado de dispositivos que são todos controlados a partir deste programa. Cada dispositivo é inteligente no sentido que será possível ligá-lo e desligá-lo e que permite registar o seu consumo energético. Chamemos, por simplificação, a estes dispositivos de *SmartDevice* e cada um deles é identificado por um código único definido pelo fabricante. Além do código cada *SmartDevice* tem também um custo de instalação que é pago independentemente se o mesmo é depois utilizado ou não.

Cada *Casa* possui vários *SmartDevice* e pode agrupá-los por divisão. Por exemplo, na sala de estar podemos ter uma coleção de dispositivos que queremos monitorizar e controlar. Deverá ser possível a um utilizador poder ligar ou desligar todos os dispositivos de uma divisão da casa.

### 1.1 Os *SmartDevice*

Existem neste momento diversos tipos de dispositivos em comercialização e que podem ser instalados nas casas. No entanto, é possível e desejável que possam ser criados mais dispositivos e adicionados às casas. Os dispositivos actuais apenas permitem uma regulação num determinado intervalo de tempo, isto é quando são ligados apenas podem ser desligados, na melhor das hipóteses, no dia seguinte. Neste momento temos disponíveis:

- lâmpadas inteligentes, as *SmartBulb*, que podem ser ligadas com uma determinada tonalidade (Neutral, Warm e Cold) e que possuem, cada uma delas, uma dimensão (em cms) e informação de consumo diário. O consumo de energia de uma destas lâmpadas é dado por uma fórmula (a conceber por cada grupo) e que é função de uma valor fixo + factor em função do tipo de luz que está a ser emitida.
- colunas inteligentes, as *SmartSpeaker*, que definem um determinado volume e uma informação de qual é a rádio online que estão a tocar, bem assim como qual é a marca de cada coluna. O consumo diário de cada coluna é também função do consumo diário de colunas daquela marca + factor em função do volume em que a coluna está a tocar.
- câmaras de vigilância, as *SmartCamera*, que determinam a resolução em que operam e o tamanho do ficheiro em que guardam os eventos registados. O consumo energético é função do tamanho do ficheiro que geram \* a resolução da imagem.

## 1.2 As casas

Cada casa possui uma colecção (no sentido lato e necessariamente de estrutura de dados) de *SmartDevice* e tem também a informação de quais as divisões da casa (sala, cozinha, quarto, etc.) e guarda também a informação de que dispositivos é que estão em cada divisão. Uma casa possui também informação de quem é o seu proprietário, nomeadamente o seu nome e o seu número de identificação fiscal (NIF).

Para uma casa deverá ser possível:

- ligar e desligar todos os dispositivos de uma divisão
- ligar e desligar um dispositivo específico

## 1.3 Os comercializadores de energia

Existem no mercado diversos fornecedores de energia e cada um deles tem uma definição de preço a que vende a energia diferente, de acordo com os seus critérios comerciais. Cada casa tem em determinado momento um contrato com uma empresa que seja um comercializador de energia, não podendo existir casas que possam estar sem fornecedor de energia associado.

Os diversos fornecedores de energia tem liberdade na construção da fórmula que determina o valor do custo diário a que vendem a energia, mas tem de respeitar algumas premissas, a saber:

- está definido no arranque do sistema qual é o valor base do custo diário do kwh de energia
- está definido no arranque do sistema qual é o factor multiplicativo dos impostos

Fora estas premissas os comercializadores podem criar a fórmula de cálculo do preço. Um exemplo seria algo como  $PrecoDiaPorDispositivo = (ValorBase * ConsumoDispositivo * (1 + Imposto)) * 0.9$ .

Outro exemplo seria  $PrecoDiaPorDispositivo = numeroDispositivos > 10 ? (ValorBase * ConsumoDispositivo * (1 + Imposto)) * 0.9 : (ValorBase * ConsumoDispositivo * (1 + Imposto)) * 0.75$

## 2 O Funcionamento do programa

Pretende-se que com este programa se possa criar uma espécie de simulação do valor do consumo de energia de várias casas. Obviamente para considerar valores de consumo de energia é necessário que decorra um determinado espaço de tempo e para ter esse efeito precisamos de estabelecer determinadas regras:

1. o programa deve ter uma fase em que é permitido criar dispositivos, casas, e fornecedores de energia. Uma hipótese é esta informação ser carregada de ficheiro dispensando os utilizadores de cada vez que usam o programa estarem a fazer o setup da informação.

2. a componente de simulação deve começar já com uma configuração em que para cada casa existem dispositivos ligados, outros desligados, e existe um contrato estabelecido com um fornecedor.
3. o programa deve pedir ao utilizador que determine o avançar do tempo. Este avançar do tempo faz-se em dias e o programa deve permitir **ir para uma determinada data**. Nessa altura calcula-se o consumo de cada casa assumindo que durante esses dias os dispositivos estiveram sempre no mesmo estado (ligados ou desligados). Se estiveram ligados, sabe-se o consumo diário do dispositivo e aplica-se a fórmula de cálculo.
4. Quando se vai para uma determinada data e se calcula o custo faz-se a emissão para cada casa da factura respectiva indicando o período, o consumo e o custo.
5. As casas podem, a qualquer altura solicitar a mudança de fornecedor de energia, mas essas acções só se materializam quando se faz o **avançar do tempo**. Esse evento permite além de fazer cálculos mudar a informação que seja necessário. Estes pedidos terão de ser guardados e executados após a geração das facturas, começando nessa altura um novo ciclo de contagem.
6. As casas podem também optar por ligar e desligar dispositivos, mas essas acções também só se executam quando se fecha um período de simulação (quando se muda de data).
7. Os comercializadores podem também alterar os valores, mas mais uma vez isso só produz efeito no próximo ciclo de facturação.

## 3 Requisitos

O trabalho proposto tem vários níveis de requisitos, desde os mais básicos até aos mais complexos. Para que os vários níveis de requisitos identificados sejam considerados cumpridos o programa deve respeitar os princípios da programação por objectos, ser robusto no tratamento de erros bem como gravar, e também carregar, o estado do programa para um ficheiro de objectos. Deverá possuir uma camada de interacção com o utilizador, que se aconselha ser feita por menús em modo texto, e que permita aceder às funcionalidades do programa. Durante a sessão de apresentação será também avaliada a forma como o programa está construído, nomeadamente a utilização das estruturas de dados e a forma como são utilizadas e manipuladas.

### 3.1 Requisitos base de gestão das entidades

Como requisitos base pretende-se ter a capacidade de criar dispositivos, casas e fornecedores de energia. Ser capaz de ligar e desligar dispositivos e permitir avançar o tempo e calcular o consumo e gerar as respectivas facturas.

Nota máxima: 12 valores

### 3.2 Efectuar estatísticas sobre o estado do programa

O programa deverá permitir saber:

1. qual é a casa que mais gastou naquele período
2. qual o comercializador com maior volume de facturação
3. listar as facturas emitidas por um comercializador
4. dar uma ordenação dos maiores consumidores de energia durante um período a determinar

Nota máxima: 16 valores

### 3.3 Alterar os operadores e os dispositivos de uma casa

O programa deverá permitir satisfazer os requisitos identificados em 2 com os números 5, 6 e 7, permitindo a alteração do estado das diversas entidades.

Nota máxima: 18 valores

### 3.4 Automatizar a simulação

A simulação sugerida até agora é feita por intervenção do utilizador sempre que este decide mudar explicitamente a data (para uma data no futuro). O sistema poderia, no entanto, admitir que se pudessem colocar eventos mais detalhados e possibilitando inclusive o ligar e desligar dispositivos, fazendo os respectivos cálculos e gerando a informação necessária. Para tal deveria ler, por exemplo um ficheiro, com informação relativa *data*, *dispositivo*, *accao* e executar esses comandos.

Por exemplo:

```
2022.04.10, casa1, dispositivo1, setOn
2022.04.10, casa2, dispositivo2, setOff
2022.04.11, casa2, fornecedorX --> muda de fornecedor
2022.04.12, fornecedorY, alteraValorDesconto, novoValor
```

Este ficheiro pode ser construído consoante cada grupo e pode até incluir além do dia informação sobre a hora em que ocorre a mudança. Estes requisitos representam um grau de complexidade acrescido e destinam-se a distinguir os trabalhos que sejam excelentes.

Nota máxima: 20 valores

Obviamente que a nota a atribuir, em cada um dos escalões, irá reflectir a estruturação da solução que deverá respeitar as normas da programação orientada aos objectos, cf aulas teóricas, nomeadamente o encapsulamento, a abstração de implementação e a capacidade de a aplicação evoluir de forma controlada (eg: através da incorporação de novos tipos de dispositivos, comercializadores, alteração das fórmulas, etc.). Não serão considerados como merecedores de nota positiva trabalhos que não permitam a criação e visualização da informação relativa às entidades do programa.

## 4 Relatório

O relatório deve descrever o trabalho realizado para desenvolver a aplicação solicitada. No mínimo, devem ser abordados os seguintes pontos:

- Capa com identificação da Unidade Curricular e do grupo (nome, número e fotografia).
- Descrição da arquitectura de classes utilizada (classes, atributos, etc.) e das decisões que foram tomadas na sua definição. Deverá ser entregue um **Diagrama de Classes** com a arquitectura de classes que suporta o programa desenvolvido.
- Descrição da aplicação desenvolvida (ilustração das funcionalidades).

## 5 Salvaguarda do estado da aplicação

O programa deve permitir que em qualquer momento se possa guardar em ficheiro a informação existente em memória sobre a informação relevante das entidades. A gravação deve ser feita de forma a permitir que o estado que foi gravado seja recuperado novamente. Na altura da entrega do projecto deve ser também entregue um estado (guardado em ficheiro) que possa ser carregado durante a apresentação.

## 6 Cronograma

1. Entrega da composição do grupo.

**Data Limite:** *22 de Abril de 2022* (esta fase é eliminatória, isto é, os grupos que não registarem a constituição não poderão submeter o projecto final).

2. Entrega final de código, ficheiro com os dados a carregar durante a apresentação e relatório de projecto (feita por via electrónica no elearning)

**Data Entrega:** de *7 de maio de 2022* a *21 de maio de 2022* (podem gerir o momento da entrega consoante for mais conveniente e podem entregar várias vezes o projecto, sendo que será apenas avaliada a última entrega)

3. Apresentação do projecto

**Semana de:** *2, 3 e 4 de junho - data acordada com a direcção de curso*