

Programação Orientada por Objetos

Smart Home



Ano Letivo: 2017/2018

Época Normal e de Recurso

2ª Fase

Índice

Índice.....	2
Introdução.....	3
Objetivos	4
Requisitos Funcionais	4
Sugestões de Desenvolvimento	Error! Bookmark not defined.
Regras de Desenvolvimento e Entrega do Projeto	5
Fases de desenvolvimento e entrega.....	5
Implementação e codificação	5
Entrega do projeto	5
Regras e Critérios de Avaliação do Projeto	7
Regras de Avaliação	7
Critérios de Avaliação.....	7
Resumo das Datas Importantes	8
Entrega da 2ª fase.....	8
Avaliações da 2ª fase	8

Introdução

O objetivo deste projeto é desenvolver, utilizando a linguagem Java e a Programação Orientada por Objetos (POO), uma aplicação de domótica. Nesta aplicação, existe uma consola central que é responsável por controlar todos os equipamentos instalados numa residência. Esta consola recebe informação de diversos sensores e permite acionar automaticamente alguns elementos como, por exemplo, o ar condicionado, a luz, câmaras, alarmes, etc. O utilizador interage com a consola, podendo também ele comandar, manualmente, alguns desses elementos.

O projeto será desenvolvido em duas fases. A primeira fase consiste na modelação e implementação do conjunto de classes que permitam representar a lógica da aplicação; a segunda fase será dedicada à criação da interface gráfica com o utilizador, em JavaFX.

Tenha particular atenção ao uso do paradigma de POO na modelação da aplicação, i.e., ao correto uso dos conceitos de encapsulamento, herança, classes abstratas, polimorfismo, interfaces, maximização da coesão (responsabilidade única), minimização do acoplamento, desenho orientado por responsabilidades, etc. Uma modelação bem pensada facilitará todo o processo de desenvolvimento e manutenção.

O presente documento versa agora sobre a segunda fase do projeto.

Objetivos

Na segunda fase do projeto deverá desenvolver uma interface gráfica, em JavaFx, que permita que o utilizador interaja com o sistema concebido na primeira fase. De uma forma genérica, a interface gráfica deve permitir introduzir a informação de instalação do sistema e definir os valores dos sensores de forma a se poder visualizar a reação dos atuadores.

Requisitos Funcionais

RF1 – Ao iniciar o sistema, o utilizador deve poder fornecer a informação de instalação, nomeadamente: os dados do cliente, as divisões, os módulos, sensores e atuadores e a distribuição dos diversos equipamentos pelas divisões.

- Utilize janelas de diálogo para receber os dados do utilizador.

RF2 – Ao ser encerrada a aplicação, deve ser perguntado ao utilizador se quer gravar a configuração existente (dados do cliente, divisões, módulos, sensores e atuadores instalados, etc.).

- Recomenda-se, neste caso, a utilização da serialização de objetos para a gravação dos dados.

RF3 – Em alternativa a fornecer os dados de instalação, o utilizador deve poder selecionar uma das configurações existentes gravadas anteriormente ou uma configuração inicial automática que pode ser feita, em código, diretamente na aplicação. A configuração automática deve criar uma instalação mínima com uma consola central, três divisões e os três módulos da consola central (ou os que tiverem sido implementados). Deve ainda, criar os sensores e atuadores suficientes para testar as funcionalidades implementadas.

RF4 – Durante a execução da aplicação, o utilizador deve poder ver a informação da instalação.

- Pode utilizar uma janela de diálogo para esta visualização.

RF5 – Durante a execução da aplicação, deve ser possível visualizar as divisões com os diversos sensores e atuadores.

- Pode optar por mostrar todas as divisões ou por mostrar apenas uma divisão permitindo ao utilizador escolher a divisão que quer visualizar.

RF6 – Durante a execução da aplicação o utilizador deve poder alterar os valores dos sensores para que os módulos da consola central consigam atuar em conformidade, agindo sobre os atuadores

- Por exemplo: ao clicar na representação de um sensor, será aberta uma *janela de diálogo* que permitir alterar os valores do sensor. Em alternativa, pode ser colocado na visualização da divisão, um icon ou um desenho do sensor juntamente com um controlo que permita fazer diretamente essa alteração sem necessidade de abrir uma janela. É o caso, por exemplo, da utilização de um slider para sensores que registam valores numéricos.

RF7 – Na visualização de sensores e atuadores deve ser incluído a visualização dos seus estados.

- Por exemplo, um ar condicionado pode incluir um número com a temperatura para a qual está programado e a indicação se está ligado ou não que podia ser feita com um círculo a verde ou a vermelho respetivamente.

RF8 – Deve ser possível visualizar o resultado da atuação dos módulos sobre os atuadores.

Uma solução pode ser a utilização de um botão que quando acionado leva a que seja efetuada a leitura do estado dos sensores, originando de seguida a atuação dos módulos que modificam o estado dos atuadores. Neste caso, os valores do sistema são atualizados e os atuadores devem reproduzir visualmente o seu estado com as alterações efetuadas (ex: sirene em estado de alerta, tomada ligada, intensidade da luz da lâmpada, fotos ou vídeos das camaras fotográficas ou de vídeo, etc);

Regras de Desenvolvimento e Entrega do Projeto

Fases de desenvolvimento e entrega

O projeto está dividido em 2 fases, com a cotação distribuída da seguinte forma:

- Fase I – 60% da avaliação final
- Fase II – 40% da avaliação final

Conforme referido na ficha da disciplina, poderá alternativamente entregar o projeto numa só data, na época de recurso (sem possibilidade de incorporar componente de avaliação contínua).

Implementação e codificação

A segunda fase do programa deve utilizar a tecnologia JavaFX, colocando em prática os conceitos fundamentais do paradigma de Programação Orientada por Objetos adquiridos anteriormente.

Em relação às regras de codificação, siga as convenções adotadas normalmente para a linguagem Java:

- A notação *camelCase* para o nome das variáveis locais e identificadores de atributos e métodos;
- A notação *PascalCase* para os nomes das classes e interfaces;
- Utilização de maiúsculas para os nomes das constantes e dos valores enumerados;
- Não utilize o símbolo ‘_’ nos identificadores (exceto nas constantes), nem abreviaturas.

É necessário que o projeto cumpra o que é pedido no seu enunciado, sendo deixado ao critério do programador qualquer pormenor de implementação que não seja referido, o qual deverá ser devidamente documentado.

Entrega do projeto

- O projeto é entregue em duas fases:
 - Uma primeira fase (**até às 23:55:00 do dia 16 de maio de 2018**) com a implementação da lógica da aplicação (descrita no presente documento);
 - A segunda fase (**até às 23:55:00 do dia 26 de junho de 2018**) com a parte gráfica.
- O projeto deverá ser entregue até à data limite especificada por **via exclusivamente eletrónica utilizando a área dos trabalhos no Moodle da unidade curricular**. Todos os ficheiros que compõem o projeto deverão estar guardados num único ficheiro compactado em **formato ZIP**. Em caso de dificuldades no acesso à plataforma Moodle, o envio dos ficheiros

poderá ser feito por correio eletrónico para o respetivo docente de laboratório, dentro do prazo acima indicado.

- **Não serão aceites quaisquer projetos entregues fora do prazo!**
- Todos os materiais do projeto devem ser devidamente identificados com nome, número e endereço de correio eletrónico dos alunos.

Os materiais do projeto deverão incluir:

- Um Manual Técnico onde conste uma breve descrição do programa, incluindo a explicação das classes/interfaces implementadas, principais atributos e métodos e suas relações.
- A documentação do programa em JavaDoc (não converta o documento gerado automaticamente em HTML para DOC!).
- O código fonte do programa na forma de projeto em *NetBeans*, com um *main* de testes a funcionar e com todas as funcionalidades implementadas.
- Todos os ficheiros que compõem o projeto deverão estar guardados num único ficheiro compactado em formato ZIP cujo nome deverá ter a seguinte nomenclatura: `<curso>_<numAluno1>_<numAluno2>.zip`.

Regras e Critérios de Avaliação do Projeto

Regras de Avaliação

A avaliação do projeto está sujeita às seguintes regras:

- Caso o aluno falte ao momento de supervisão, terá essa componente avaliada com zero valores.
- **Não serão aceites quaisquer projetos entregues fora do prazo!**
- A classificação do programa terá em conta a qualidade da programação (fatores de qualidade do software), a estrutura do código criado segundo os princípios da Programação Orientada por Objetos, tendo em conta conceitos como a coesão de classes e métodos, o grau de acoplamento entre classes e o desenho de classes orientado pela responsabilidade, e a utilização/conhecimento da linguagem Java.
- Serão premiadas a facilidade de utilização, a apresentação, a imaginação e a criatividade.
- O projeto terá uma componente de avaliação oral obrigatória com classificação individual dos elementos do grupo.
- Os alunos que não comparecerem à discussão serão classificados com zero na fase respetiva. Nesta discussão será apurada a capacidade do aluno de produzir o código apresentado. Nos casos em que essa capacidade não for demonstrada, a nota atribuída será zero.
- A avaliação oral é realizada pelo respetivo professor de laboratório e irá ser feita uma marcação prévia para cada grupo de trabalho.
- Todos os projetos serão submetidos a um sistema automático de deteção de cópias. Os projetos que forem identificados como possíveis cópias, e verificando-se serem realmente cópias, serão anulados.

Critérios de Avaliação

A segunda fase do projeto será avaliada segundo os seguintes critérios:

Componentes e funcionalidades associadas	55%
Dados da instalação (RF1 e RF3)	5%
Utilização e gravação de configurações (RF2, RF3)	5%
Visualização da configuração (RF4)	5%
Visualização das divisões com os sensores e atuadores (RF5)	15%
Alteração dos valores dos sensores (RF6)	10%
Visualização do estado dos sensores e atuadores (RF7)	10%
Simulação da operação dos módulos (RF8)	5%

Implementação	25%
Utilização correta dos controlos	10%
Usabilidade da aplicação	10%
Bom estilo (nomes, comentários, indentação)	5%

Documentação	10%
JavaDOC	5%
Manual técnico	5%

Avaliação qualitativa	10%
------------------------------	------------

Resumo das Datas Importantes

Entrega da 2ª fase

A entrega da 2ª fase do projeto será até às **23:55 de terça, dia 26 de junho de 2018**.

Avaliações da 2ª fase

As avaliações da segunda fase do projeto serão realizadas de acordo com as disponibilidades do docente e dos alunos e combinadas entre os mesmos.