

# Projeto de Programação Orientada para Objetos 2014 - 2015

## Simulador de consumo eléctrico

### 1. Introdução

A recente criação de aparelhos domésticos que comunicam juntando-se à [Internet of Things](#) (IoT) e de contadores de energia "inteligentes" permite construir aplicações a pensar nestes sistemas. O trabalho proposto é uma simulação de um ambiente doméstico em que há vários aparelhos a contribuir para o consumo de energia. Queremos por um lado parametrizar a simulação (quais os aparelhos ligados, onde e por que períodos) e depois queremos ver o gráfico resultante do consumo por ramal.

### 2. Funcionamento geral (versão intercalar)

Deve implementar um sistema de simulação de consumo eléctrico, com as seguintes funcionalidades.

1. Inicializar a grelha eléctrica de uma casa, composta por linhas eléctricas e tomadas;
2. Adicionar aparelhos eléctricos a cada uma das linhas;
3. Correr iterativamente desde o momento zero até atingir o ciclo correspondente ao tempo de simulação final (para já pode ser uma constante). Cada iteração do ciclo simulará uma unidade de tempo a passar;
4. O resultado da execução do programa de teste inicial (versão 1.0 do MainIntercalar) deverá ser:

```
t = 0
cozinha 0.0W
quartos 0.0W
sala 0.0W
```

```
t = 250
cozinha 20.0W
quartos 0.0W
sala 0.0W
```

```
t = 500
cozinha 50.0W
quartos 40.0W
sala 0.0W
```

```
t = 750
cozinha 20.0W
quartos 40.0W
sala 0.0W
```

```
t = 1000
cozinha 20.0W
quartos 40.0W
sala 0.0W
```

### 3. Formato dos ficheiros de configuração

Apenas serão usados na versão final ...

### 4. Requisitos de implementação

1. Deve respeitar o desenho na Figura 1. Pode evidentemente (e deve) juntar classes ao desenho.
2. Todos os testes (informais ou JUnits) fornecidos até 16 de Março deverão estar a funcionar na entrega intercalar.
3. As classes devem estar adequadamente distribuídas por *packages*, de modo a serem facilmente transferidas para outro projeto;

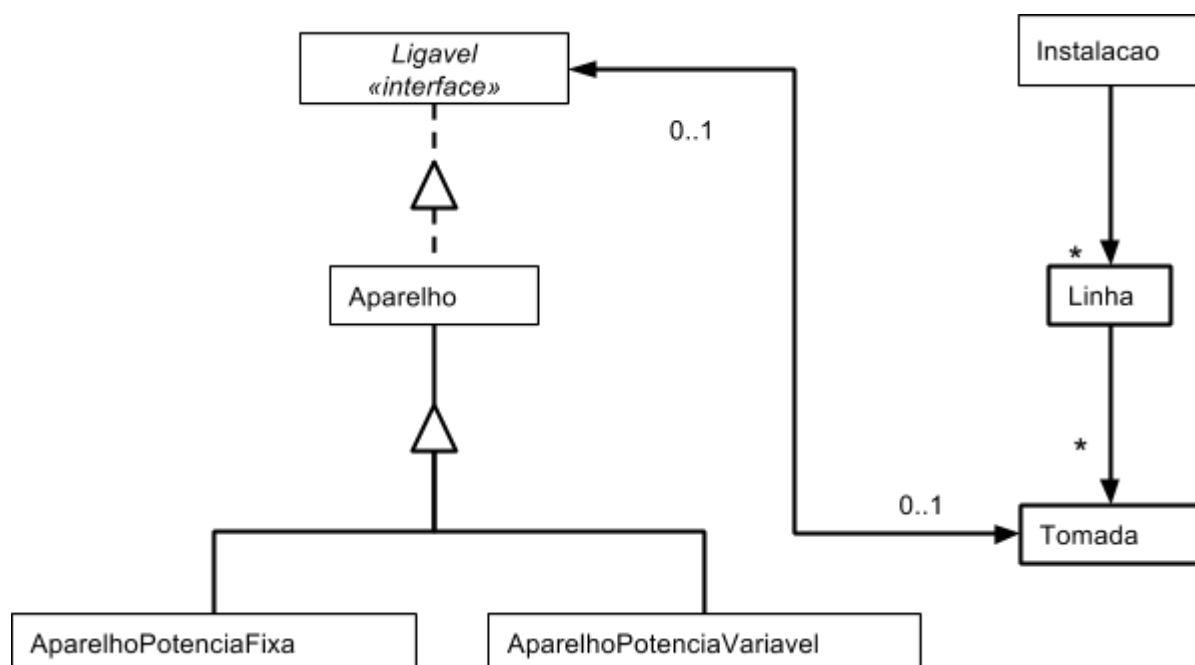


Figura 1. Desenho parcial, de alto nível, das relações entre as principais classes da aplicação. Uma das classes **Linha** ou **Tomada** (mas não ambas) pode não existir desde que sejam mantidas as funcionalidades que lhe dizem respeito.

### 5. Requisitos de entrega

O não cumprimento dos requisitos numa das entregas pode implicar a eliminação do aluno / grupo do processo de avaliação, só podendo voltar a tentar obter aprovação à disciplina em Época Especial, se admissível (entregando novo projeto) ou no próximo ano letivo.

A decisão sobre o cumprimento ou não dos requisitos é do docente que acompanhou o trabalho, tal como no caso de uma entrega, intercalar ou final, de um projeto real, um cliente pode decidir que as condições do contrato não foram cumpridas e terminar o projeto abandonando a encomenda feita.

Versão intercalar:

1. Deve entregar a versão intercalar no e-learning, na pasta "Entrega Intercalar do Trabalho Final 2015 - <Nome do Docente>" e por mail para o docente que acompanhou o trabalho. A data limite de entrega da versão intercalar do trabalho é até às 08:00 de 13 de Abril de 2015 (a avaliação desta entrega será presencial na aula prática, numa das semanas seguintes).
2. Se nenhum docente acompanhou o trabalho por não frequentar habitualmente as aulas, deve entregar, em "Conteúdos da Unidade Curricular" na pasta "Entrega Intercalar do Trabalho Final 2015 - Sem Docente" e enviar cópia por mail para o coordenador da UC ([luis.nunes@iscte.pt](mailto:luis.nunes@iscte.pt));
3. Não deve usar símbolos acentuados no código do trabalho, nem mesmo nos comentários.
4. Deve entregar apenas um pacote que possa ser automaticamente importado para o Eclipse com Import/Existing Projects into workspace/ Archive File (teste esta importação num computador diferente daquele onde fez o trabalho). Deve exportar usando Export / Archive File. Teste o pacote, importando-o para um *workspace* novo;
1. O nome do projecto e do zip deve incluir o nome e numero dos membros do grupo (use Refactor/Rename para mudar o nome do projecto).
2. Caso não consiga enviar por mail o zip, altere a extensão do ficheiro para "xip". Não use um compactador que gera ficheiros .rar.

## 6. Avaliação

1. O trabalho pode ser feito individualmente ou aos pares (grupos de dois elementos). Não são autorizados grupos de mais do que 2 elementos;
2. Os trabalhos serão verificados em termos de plágio e, além de obviamente serem anulados os trabalhos plagiados (o que não permitirá fazer a UC neste ano letivo), serão feitas queixas ao Conselho Pedagógico sempre que os resultados da análise indiquem claramente que houve plágio de modo a que a penalização seja efetiva;
3. Os trabalhos serão avaliados pelo cumprimento dos requisitos funcionais e de implementação. Deverá apontar para que todos os requisitos sejam cumpridos, tal como num contrato normal com um cliente.
4. Todos os trabalhos deverão ter uma discussão obrigatória, embora grupos e/ou alunos específicos possam ser dispensados caso o acompanhamento do trabalho nas aulas dê ao docente segurança suficiente para comprovar que o aluno é responsável pelo trabalho;
5. Trabalhos que não cumpram vários dos requisitos não permitirão ao aluno o acesso à prova da disciplina e por isso implicam a reprovação à disciplina.
6. Requisitos da entrega intercalar:
  - o Respeitar o desenho de classes
  - o Passar nos testes informais e JUnit publicados até 16 de Março.
7. As normas quanto ao cumprimentos dos requisitos de entrega serão rígidas, o programa deve cumprir TODOS os requisitos.