

**Nome:**

**Número:**

MESTRADO INTEGRADO EM ENGENHARIA INFORMÁTICA  
UNIVERSIDADE DO MINHO

# Sistemas Distribuídos

## Teste

9 de Janeiro de 2017

Duração: 2h00m

# I

1 Distinga, em termos de objetivo e forma de utilização, as primitivas de *lock/unlock* e *wait/notify*.

[illegible]

2 Das arquiteturas de sistemas distribuídos que estudou qual ou quais se adequariam melhor a um sistema de suporte a redes sociais?

[illegible]

**3** Qual a relevância do sistema operativo na resolução do problema de exclusão mútua no modelo de memória partilhada e no modelo de passagem de mensagens?

[illegible]<sup>1</sup>Cotação — 10+10

## II

Considere um serviço simplificado de controlo de aquecimento numa casa (exemplo de *internet of things*) onde os equipamentos se ligam por TCP/IP a um servidor. Existem os seguintes dispositivos: Um termómetro que se liga para indicar a **temperatura** actual da casa; Um controlo de temperatura alvo que se liga para indica o **limiar** de activação da caldeira; Um relé que controla a caldeira e que indica o seu estado actual (off = false ou on = true) e aguarda indicação de mudança de estado via o método **on\_off**. Os métodos **temperatura** e **limiar** devem retornar imediatamente, mas o método **on\_off** deve bloquear caso o estado fornecido seja idêntico ao estado actual do sistema. Por exemplo, se a temperatura actual for de 16° e o limiar actual for de 19° então o sistema sabe que a caldeira deve estar ligada para aquecer a casa; se o relé da caldeira invocar `on_off(true)`, indicando já estar ligado, então o método só deve retornar (indicando **false**) quando a temperatura ultrapassar os 19°.

```
interface Controlador {  
    temperatura(int centigrados);           // medidor indica temperatura actual  
    limiar(int centigrados);                // utilizador indica limiar de activação  
    boolean on_off(boolean estadoatual);    // caldeira pergunta se estado mudou  
}
```

- 1 Apresente uma classe que implemente o interface Controlador tendo em conta que os seus métodos poderão ser invocados num ambiente multi-threaded.
- 2 Implemente um servidor em rede usando sockets TCP que disponibilize os métodos da classe desenvolvida na pergunta anterior. O servidor deverá suportar a conexão de múltiplos clientes em simultâneo.