

Parte A (12 valores)

- ~~1.~~ Explique o que é um sistema distribuído e indique dois dos aspectos que têm que ser ponderados na sua concepção.
- ~~2.~~ Caracterize o modelo cliente-servidor e indique qual é o papel aí desempenhado por um servidor de 'proxies'.
- ~~3.~~ Descreva o modelo de programação providenciado pelo protocolo TCP, quer do lado do servidor, quer do lado do cliente.
4. Em que consiste o método de invocação remota? Será que dentro deste modelo é possível a migração de código? Como?

Parte B (8 valores)

As classes de Java apresentadas abaixo descrevem um modelo produtor-consumidor muito simples.

```
public class Producer extends Thread
{
    private CubbyHole cubbyhole;
    private int number;
    public Producer(CubbyHole c, int number)
    {
        cubbyhole = c;
        this.number = number;
    }
    public void run()
    {
        int val;
        for (int i = 0; i < 10; i++)
        { val = 10*number+i;
          cubbyhole.put (number, val);
          try
          {
              sleep((int) (Math.random() * 100));
          }
          catch (InterruptedException e) { }
        }
    }
}

public class Consumer extends Thread
{
    private CubbyHole cubbyhole;
    private int number;
    public Consumer(CubbyHole c, int number)
    {
        cubbyhole = c;
        this.number = number;
    }
    public void run()
    {
        int value = 0;
        for (int i = 0; i < 10; i++)
        { value = cubbyhole.get (number);
        }
    }
}
```

```
public class CubbyHole
{
    private int contents;
    private boolean available = false;
    public synchronized int get(int number)
    {
        while (!available)
        { try
          { wait ();
            }
          catch (InterruptedException e) { }
        }
        System.out.println("Consumidor #" + number + " consome: " +
                           contents);
        available = false;
        notify ();
        return contents;
    }
    public synchronized void put (int number, int value)
    {
        while (available)
        { try
          { wait ();
            }
          catch (InterruptedException e) { }
        }
        contents = value;
        System.out.println("Produtor #" + number + " produz: " +
                           contents);
        available = true;
        notify ();
    }
}

public class ProducerConsumer
{
    public static void main (String[] args)
    {
        CubbyHole c = new CubbyHole();
        Producer pl = new Producer(c, 1);
        Consumer cl = new Consumer(c, 1);
        pl.start();
        cl.start();
    }
}
```

- a) Que valores são impressos no ecrã do monitor vídeo quando este programa é executado?
- b) Qual é o papel da variável *available* da class *CubbyHole* no estabelecimento da sincronização? Justifique a sua resposta.
- c) Altere o programa anterior para possibilitar o lançamento de 2 processos produtores e 2 processos consumidores e ter uma capacidade de armazenamento interno de 5 valores.