

Parte A (12 valores)

(10)

Um atributo importante de um sistema distribuído é a sua transparência. Explique em que consiste e, neste contexto, distinga transparência de acesso de transparência de localização.

2. Qual é o papel desempenhado por um registry numa arquitectura distribuída que disponibiliza um conjunto de serviços na Internet? Faça um diagrama que ilustre a sua localização.

Quais são as diferenças principais entre os protocolos TCP e UDP?

Uma maneira de se efectuar a localização remota de objectos dentro do modelo de invocação remota de métodos (RMI) é através do rmiregistry. Explique em que consiste e descreva as operações que têm que ser realizadas pelo servidor e pelo cliente para que tal seja possível.

Parte B (8 valores)

(10)

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

```
public class Producer extends Thread
{
    private Fifo fifo;
    private int number;

    public Producer(Fifo c, int number)
    {
        fifo = c;
        this.number = number;
    }

    public void run()
    {
        int val;

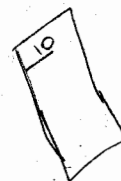
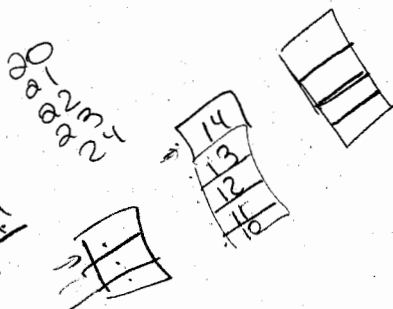
        for (int i = 0; i < 5; i++)
        {
            val = 10*number+i;
            fifo.in (number, val);
            try
            {
                sleep((int)(Math.random() * 100));
            }
            catch (InterruptedException e) { }
        }
    }
}
```

```
public class Consumer extends Thread
{
    private Fifo fifo;
    private int number;

    public Consumer(Fifo c, int number)
    {
        fifo = c;
        this.number = number;
    }

    public void run()
    {
        int value = 0;
        for (int i = 0; i < 5; i++)
        {
            value = fifo.out (number);
        }
    }
}
```

am registry 24001



Ag
Codigo 1001 ->

Adicional -> mascara
modelo

Agente -> model
aplet ->
Codigo

```

public class FiFo
{
    private int N;
    private int [] contents;
    private boolean empty;
    private boolean full;
    private int pin, pout;

    public FiFo (int N)
    {
        this.N = N;
        contents = new int [N];
        empty = true;
        full = false;
        pin = 0;
        pout = 0;
    }

    public synchronized int out (int number)
    {
        int tmp;
        while (empty)
        {
            try
            { wait(); }
            catch (InterruptedException e) { }
        }
        System.out.println("Consumidor #" + number + " consome: " +
        contents[pout]);
        tmp = contents[pout];
        pout = (pout + 1) % N;
        if (pin == pout) empty = true;
        if ((empty && (pout == (pin+1) % N)) full = false;
        notifyAll ();
        return tmp;
    }

    public synchronized void in (int number, int value)
    {
        while (full)
        { try
            { wait (); }
            catch (InterruptedException e) { }
        }
        contents[pin] = value;
        System.out.println("Produtor #" + number + " produz: " +
        contents[pin]);
        pin = (pin + 1) % N;
        if (pin == pout) full = true;
        if ((full && (pin == (pout+1) % N)) empty = false;
        notifyAll ();
    }
}

public class ProducerConsumer
{
    public static void main (String[] args)
    {
        FiFo c = new FiFo (3);
        Producer p1 = new Producer(c, 1); Producer p2 = new Producer(c, 2);
        Consumer c1 = new Consumer(c, 1); Consumer c2 = new Consumer(c, 2);
        p1.start(); p2.start();
        c1.start(); c2.start();
    }
}

```

-
- a) Que valores são impressos no écran do monitor vídeo quando este programa é executado? Dê um exemplo.
 - b) Qual é o papel das variáveis *empty* e *full* da class *FiFo* no estabelecimento da sincronização? Serão sempre necessárias duas variáveis? Justifique a sua resposta.
 - c) Altere o programa anterior para criar uma memória com capacidade de armazenamento interno de 5 valores e que permita que sejam retirados os valores por ordem inversa da sua introdução.