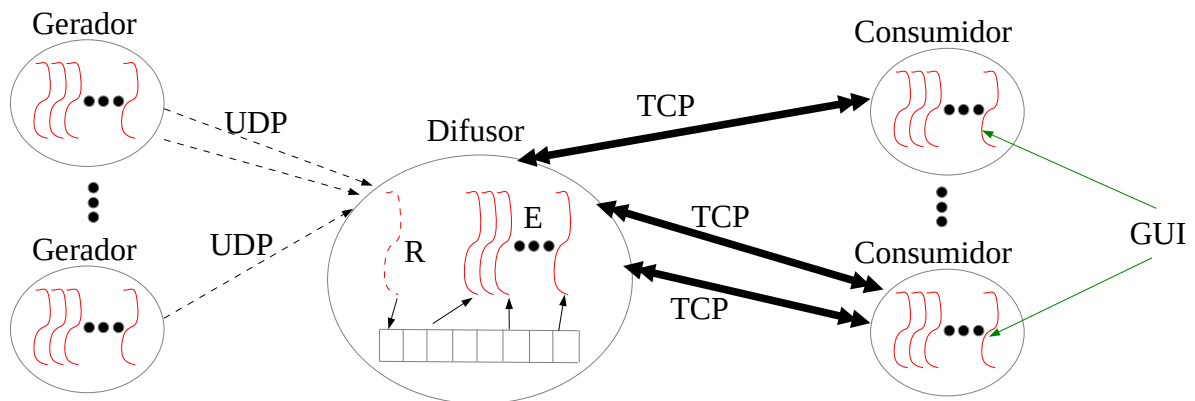


Trabalho de implementação de Aplicações Distribuídas

Simulador Pub/Sub Difusor de informações



O trabalho será composto de três aplicações: Gerador, Difusor e Consumidor. O Gerador deve gerar informações a serem difundidas pelo Difusor. Os Consumidores se inscrevem no Difusor para receber o tipo de informação desejada.

Informação: é um objeto gerado nos geradores. Esse objeto deve possuir três (3) atributos: seq, tipo e valor. Usar os tipos abaixo:

```
int seq;  
int tipo;  
int valor;  
public Informacao desempacota(String msg){}
```

Tipos de Informação: 1	Esportes
2	Novidades da Internet
3	Eletrônicos
4	Política
5	Negócios
6	Viagens

Gerador:

Cada gerador pode gerar mais de um tipo de informação, criando uma thread para cada tipo de informação desejada. Deve-se informar a quantidade de geradores a serem criados. Cada *thread* deve comunicar a informação por meio de socket UDP ao Difusor. Não é necessário interface gráfica e nem entrada pelo usuário (pode ficar no código as opções).

Cada thread geradora deve receber: tipo da informação, valor máximo (max), valor mínimo (min), tempo máximo (tmax) e tempo mínimo (tmin). Deve-se gerar um valor aleatório, entre min e max e enviar a informação ao difusor. O valor de seq não precisa ser preenchido pois será atualizado no Difusor. Gerar outro valor aleatório 'dorme', entre tmin e tmax e adormecer a thread pela quantidade de 'dorme' em ms.

Difusor:

O difusor vai receber as informações dos geradores e solicitações de conexão dos consumidores. Ele vai passar a cada consumidor as informações referentes ao(s) tipo(s) escolhido(s). A aplicação deve possuir pelo menos duas *threads*: uma para receber as informações

dos geradores e colocar numa fila e outra para aguardar a conexão dos consumidores.

A *thread* que recebe a informação dos geradores deve colocar um número único (seq) na informação e em seguida inserir um objeto informação na fila. Esse número deve iniciar em zero (0) e ser incrementado a cada nova informação recebida.

A *thread* (pode ser no fluxo principal da aplicação) que aguarda as conexões dos consumidores. A cada conexão dos consumidores (TCP) deve-se criar uma *thread* para atender as necessidades desse consumido e aguardar novos pedido de conexão.

Cada *thread* criada para atender ao consumidor deve receber (TCP) o tipo de informação desejado. Após receber, deve enviar (TCP) ao consumidor toda a informação contida na fila daquele tipo. A cada nova informação daquele tipo inserida na fila, deve-se enviar ao consumidor, com o cuidado de não enviar mais de uma vez a mesma informação (seq).

Consumidor:

O consumidor pode possuir uma interface gráfica (ou texto) para mostrar as informações recebidas. Deve possuir pelo menos duas *threads*: uma da interface gráfica (GUI) e outras das informações desejadas. Um consumidor pode consumir um (1) tipo de informação diferentes.

O consumidor deve enviar o tipo de informação desejada ao difusor (TCP). Em seguida, deve-se receber (TCP) as informações do difusor e mostrar na interface gráfica (ou texto). A interface deve possuir uma opção de parar de receber informações, onde deve-se fechar a conexão e encerrar a *thread*.

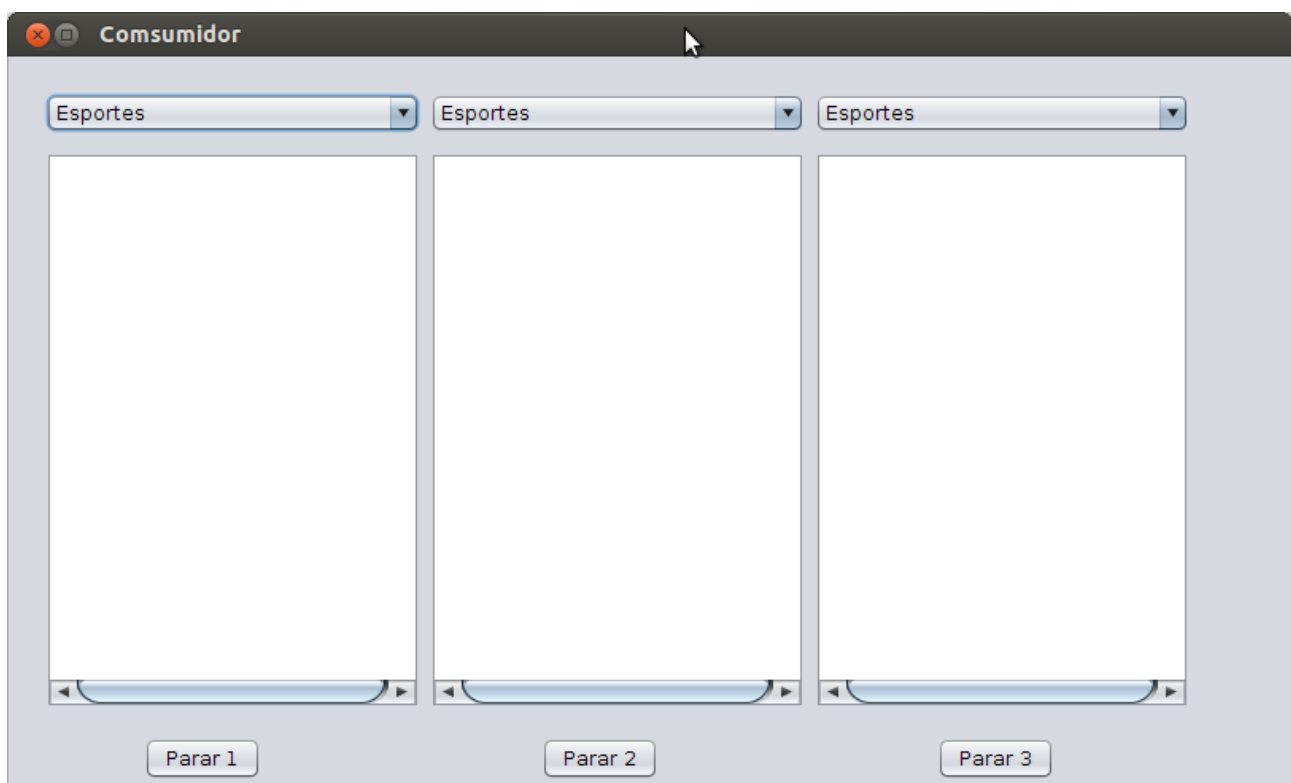


Figura 1 – Exemplo de consumidor com até 3 tipos de informação solicitadas.

PS. Os Ips e portas podem ser colocados fixos nas aplicações. Sugiro que seja utilizada mais de uma máquina para os testes.