

Nome: _____ Matrícula: _____

1. Considere uma aplicação distribuída, em que um objeto cliente se comunica com um objeto servidor via um canal bidirecional de comunicação. Este canal segue a interface abaixo:

```
public interface Channel {
    void send(Object o) throws RemoteException;
    Object receive() throws RemoteException;
    void connect(String hostName) throws RemoteException;
    void disconnect() throws RemoteException;}

```

- (a) Diferentes implementações desta interface provêm diferentes funcionalidades. Por exemplo, existe uma implementação, `CryptoChannel` que envia os dados criptografados. Um outra implementação de `Channel`, chamada `ZipChannel`, envia os dados compactados. Como construir uma implementação que envia os dados compactados e também criptografados re-usando (e possivelmente modificando) estas duas implementações?

- (b) Descreva uma forma sistemática de compormos cadeias de canais. Isto é, dados canais c_1, c_2, \dots, c_n , em que cada canal $c_i, 1 \leq i \leq n$ provê uma funcionalidade possivelmente diferente, gostaríamos de produzir um canal $c_{1,2,\dots,n}$ que agrupe todas estas funcionalidades. Dica: o padrão de projetos *Decorator* pode ser útil aqui. Claro, esta não é a única forma de implementarmos cadeias de canais, mas talvez você queirida dar uma lida neste padrão mesmo assim...