

```

public class ArchivoTexto {
    int size;
    string text;
    string owner;

    public ArchivoTexto(int size, string text, string owner) {
        this.size = size;
        this.text = text;
        this.owner = owner;
    }
}

public interface IServidor {
    public void conectar(string user, string password);
    public void enviarArchivo(ArchivoTexto archivo);
    public void desconectar(string user);
    public List<ArchivoTexto> sincronizarArchivos();
    public void recibirArchivo(ArchivoTexto archivo);
    public int tasaDeTransferencia();
    public void realizarBackup();
    public void programarReincio();
}

public class ServidorFTP: IServidor {
    public void conectar(string user, string password){}
    public void enviarArchivo(ArchivoTexto archivo){}
    public void desconectar(string user){}
    public List<ArchivoTexto> sincronizarArchivos(){}
    public void recibirArchivo(ArchivoTexto archivo){}
    public int tasaDeTransferencia(){}
    public void realizarBackup(){};
    public void programarReincio(){};
}

```

```
public class ClienteTransfer {
    public ServidorFTP servidor;
    public String user;
    public String password;
    public List<ArchivoTexto> archivos;
    private List<ArchivoTexto> papeleraReciclaje;

    public void conectar(){
        this.servidor.conectar(this.user, this.password);
    }

    public void desconectar(){
        this.servidor.desconectar(this.user);
    }

    public void enviarArchivos(){
        foreach (ArchivoTexto archivo in archivos) {
            this.servidor.enviarArchivo(archivo);
        }
    }

    public void recibirArchivo(ArchivoTexto archivo){
        this.servidor.recibirArchivo(archivo);
        this.archivos.Add(archivo);
    }

    public void borrarArchivo(ArchivoTexto archivo){
        this.archivos.Remove(archivo);
        this.papeleraReciclaje.Add(archivo);
    }

    public void eliminarDePapelera(ArchivoTexto archivo){
        this.papeleraReciclaje.Remove(archivo);
    }
}
```