java.rmi / Tipo composto usado na interacção

```
package aula1;

public class FileInfo implements java.io.Serializable
{
	private static final long serialVersionUID = -4498079336259690561L;

	public String name;
	public long length;
	public Date modified;
	public boolean isFile;

	public FileInfo( String name, long length, Date modified, boolean isFile) {
		this.name = name;
		this.length = length;
		this.modified = modified;
		this.isFile = isFile;
	}

	public String toString() {
		return "Name : " + name + …;
	}
}
```

java.rmi / Interface do Objecto Remoto

```
package aula1;

import java.rmi.* ;

public interface IFileServer extends Remote {
	/**
	 * Lista nome de ficheiros num dado directorio
	 */
	public String[] dir( String path) throws RemoteException,
	                                                                InfoNotFoundException;
	/**
	 * Devolve informacao sobre ficheiro.
	 */
	public FileInfo getFileInfo( String path, String name)
	                                    throws RemoteException, InfoNotFoundException;
}

public class InfoNotFoundException extends Exception {
	private static final long serialVersionUID = 4063033701940593855L;
	public InfoNotFoundException( String message) {
		super( message);
	}
}
```

java.rmi / Implementação do Objecto Remoto

```
package aula1;

import java.rmi.* ;
import java.rmi.server.* ;

public class FileServerImpl extends UnicastRemoteObject implements IFileServer {

    private File basePath;

    protected FileServerImpl( String pathname) throws RemoteException {
        super();
        basePath = new File( pathname);
     }

     public String[] dir(String path) throws RemoteException, InfoNotFoundException {
        File f = new File( basePath, path);
        if( f.exists())
            return f.list();
        else
            throw new InfoNotFoundException( "Directory not found :" + path);
    }
     ......

    public static void main( String args[] ) throws Exception {

        System.getProperties().put("java.security.policy", "aula1/policy.all") ;

        if( System.getSecurityManager() == null ) {
            System.setSecurityManager( new RMISecurityManager()) ;
        }
        try {  // start rmiregistry
            LocateRegistry.createRegistry( 1099);
        } catch( RemoteException e) {
                // do nothing – already started with “rmiregistry”
        }

        FileServerImpl server = new FileServerImpl();

        Naming.rebind("/myFileServer", server ) ;
        System.out.println("File server bound in registry") ;
    }
}
```

java.rmi / Invocação do Objecto Remoto

```
package aula1;

import java.rmi.* ;

public class GetFileInfo {

    public static void main(String[] args) throws Exception {
        if( args.length != 3) {
            System.out.println( "Use: java GetFileInfo server_host path name");
            System.exit(0);
        }
        String serverHost = args[0];
        String path = args[1];
        String name = args[2];

        IFileServer server;

        server =(IInfoServer) Naming.lookup("//"+serverHost + "/myFileServer") ;

        FileInfo info = server.getFileInfo(path, name);

        System.out.println( info) ;
    }
}
```