

Prática 10

1. Configurando propriedades e segurança em RMI.

- Interface Hello.java

```
import java.rmi.Remote;  
import java.rmi.RemoteException;  
  
public interface Hello extends Remote {  
    String oi() throws RemoteException;  
}
```

- Classe ImplHello.java

```
public class ImplHello implements Hello{  
  
    @Override  
    public String oi() {  
        return "Exemplo de programa RMI.";  
    }  
}
```

- Classe ServidorSHello.java

```
package rmi.servidor;  
  
import java.rmi.registry.LocateRegistry;  
import java.rmi.registry.Registry;  
import java.rmi.server.UnicastRemoteObject;  
  
public class ServidorSHello {  
  
    public static void main(String args[]) {  
  
        System.setProperty("java.security.policy", "java.policy");  
  
        if (System.getSecurityManager() == null) {  
            System.setSecurityManager(new SecurityManager());  
        }  
  
        System.setProperty("java.rmi.server.hostname", "127.0.0.1");  
    }  
}
```

```

    try {

        //criar objeto servidor
        ImplHello refObjetoRemoto = new ImplHello();

        Hello skeleton = (Hello) UnicastRemoteObject.exportObject(refObjetoRemoto, 0);

        LocateRegistry.createRegistry(20002);

        Registry registro = LocateRegistry.getRegistry(20002);

        /* O método bind é então chamado no stub do registro para vincular
        * o stub do objeto remoto ao nome "Hello" no registro.*/

        registro.bind("Hello", skeleton);

        System.err.println("Servidor pronto:");

    } catch (Exception e) {
        System.err.println("Server exception: " + e.toString());
        e.printStackTrace();
    }
}

```

- Arquivo java.policy (deve estar dentro do projeto).

```

grant {
    permission java.security.AllPermission;
};

```

- Classe Cliente.java

```

import java.rmi.registry.LocateRegistry;
import java.rmi.registry.Registry;
import java.util.Scanner;

import rmi.servidor.Hello;

public class ClienteSHello {

    private ClienteSHello() {}

    public static void main(String[] args) {

        System.setProperty("java.security.policy", "java.policy");

        if (System.getSecurityManager() == null) {
            System.setSecurityManager(new SecurityManager());
        }

        Scanner teclado = new Scanner(System.in);
    }
}

```

```
System.out.println("Informe o nome/endereço do RMIRegistry:");
String host = teclado.nextLine();

try {

    Registry registro = LocateRegistry.getRegistry(host, 20002);

    Hello stubObjRemotoCliente = (Hello) registro.lookup("Hello");

    String resposta = stubObjRemotoCliente.oi();

    System.out.println("Chamando método oi(): " + resposta);

    teclado.close();

} catch (Exception e) {
    System.err.println("Client exception: " + e.toString());
    e.printStackTrace();
}
}
```

Exercícios:

1. Crie um projeto no Eclipse, codifique, execute e observe o funcionamento dessa aplicação.
2. Para saber mais sobre as configurações de propriedades e segurança, acesse:
 - <https://docs.oracle.com/javase/8/docs/platform/rmi/spec/rmi-properties2.html>
 - <https://docs.oracle.com/javase/8/docs/api/java/rmi/RMISecurityManager.html>