

1. Configuração do pom.xml

No arquivo pom.xml, adicione as dependências necessárias para o uso do RMI:

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-
4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>com.exemplo</groupId>
  <artifactId>JavaRPC</artifactId>
  <version>1.0-SNAPSHOT</version>
  <dependencies>
    <!-- Dependência para compilar corretamente com RMI -->
    <dependency>
      <groupId>javax.annotation</groupId>
      <artifactId>javax.annotation-api</artifactId>
      <version>1.3.2</version>
    </dependency>
  </dependencies>
  <build>
    <plugins>
      <!-- Plugin para incluir dependências ao criar o jar final -->
      <plugin>
        <groupId>org.apache.maven.plugins</groupId>
        <artifactId>maven-assembly-plugin</artifactId>
        <version>3.3.0</version>
        <configuration>
          <archive>
            <manifest>
              <mainClass>com.exemplo.Server</mainClass>
            </manifest>
          </archive>
          <descriptorRefs>
            <descriptorRef>jar-with-dependencies</descriptorRef>
          </descriptorRefs>
        </configuration>
        <executions>
          <execution>
            <id>make-assembly</id>
            <phase>package</phase>
            <goals>
              <goal>single</goal>
            </goals>
          </execution>
        </executions>
      </plugin>
    </plugins>
  </build>
</project>
```

2. Implemente a Interface Remota

No diretório `src/main/java/com/ exemplo`, crie uma interface chamada `Hello.java`:

```
package com.exemplo;

import java.rmi.Remote;
import java.rmi.RemoteException;

// Interface remota que será usada para o RMI
public interface Hello extends Remote {
    String sayHello(String name) throws RemoteException;
}
```

3. Implemente a Classe Servidora

Crie a classe `HelloImpl.java`, que implementa a interface remota:

```
package com.exemplo;

import java.rmi.RemoteException;
import java.rmi.server.UnicastRemoteObject;

// Implementação da interface remota
public class HelloImpl extends UnicastRemoteObject implements Hello {

    protected HelloImpl() throws RemoteException {
        super();
    }

    @Override
    public String sayHello(String name) throws RemoteException {
        return "Olá, " + name + "! Este é um exemplo de RMI.";
    }
}
```

5. Crie o Servidor

Crie a classe `Server.java` para publicar o serviço:

```
package com.exemplo;

import java.rmi.Naming;
import java.rmi.registry.LocateRegistry;

public class Server {
    public static void main(String[] args) {
        try {
            // Cria e exporta o registro RMI na porta 1099
            LocateRegistry.createRegistry(1099);

            // Cria uma instância do serviço e o vincula ao registro
        }
    }
}
```

```

        HelloImpl hello = new HelloImpl();
        Naming.rebind("HelloService", hello);

        System.out.println("Servidor RMI pronto!");
    } catch (Exception e) {
        e.printStackTrace();
    }
}
}

```

6. Crie o Cliente

Crie a classe `Client.java` para acessar o serviço remoto:

```

package com.exemplo;

import java.rmi.Naming;

public class Client {
    public static void main(String[] args) {
        try {
            // Procura pelo serviço remoto no registro
            Hello hello = (Hello) Naming.lookup("rmi://localhost:1099/HelloService");

            // Invoca o método remoto
            String response = hello.sayHello("Usuário");
            System.out.println("Resposta do servidor: " + response);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}

```