

1. Configuração do pom.xml

```
<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-RMI</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 gerar o jar final -->
      <plugin>
        <groupId>org.apache.maven.plugins</groupId>
        <artifactId>maven-assembly-plugin</artifactId>
        <version>3.3.0</version>
        <configuration>
          <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. Implementação do Código

2.1 Interface Remota

Crie o arquivo Hello.java em src/main/java/com/exemplo:

```
package com.exemplo;
```

```
import java.rmi.Remote;
```

```
import java.rmi.RemoteException;

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

2.2 Implementação do Serviço

Crie o arquivo HelloImpl.java em src/main/java/com/exemplo:

```
package com.exemplo;

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

// Implementação do serviço remoto
public class HelloImpl extends UnicastRemoteObject implements Hello {

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

    @Override
    public String sayHello(String name) throws RemoteException {
        return "Olá, " + name + "! Bem-vindo ao Java RMI.";
    }
}
```

2.3 Servidor

Crie o arquivo Server.java em src/main/java/com/exemplo:

```
package com.exemplo;

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

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

            // Instancia o serviço e o registra
            HelloImpl hello = new HelloImpl();
            Naming.rebind("HelloService", hello);

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

2.4 Cliente

Crie o arquivo Client.java em src/main/java/com/exemplo:

```
package com.exemplo;
```

```
import java.rmi.Naming;
```

```
public class Client {  
    public static void main(String[] args) {  
        try {  
            // Busca o serviço remoto  
            Hello hello = (Hello) Naming.lookup("rmi://localhost:1099/HelloService");  
  
            // Faz a chamada ao método remoto  
            String response = hello.sayHello("Usuário");  
            System.out.println("Resposta do servidor: " + response);  
        } catch (Exception e) {  
            e.printStackTrace();  
        }  
    }  
}
```