



UNIVERSIDADE
LUSÓFONA

Computação Distribuída | LEI | 15/12/24

Sistema de Marcação de Consultas Clínicas

Trabalho realizado por:

Daniel Nascimento, a22208338

Ricardo Gonçalves, a22208676

Ricardo Piedade, a22207722

www.ulusofona.pt

1. Introdução

Este relatório descreve todo o desenvolvimento do projeto de Marcação de Consultas Clínicas, realizado no âmbito da disciplina de Computação Distribuída. O objetivo deste projeto é criar um sistema que seja capaz de gerir a marcação e gestão de consultas em diversas clínicas, utilizando uma arquitetura baseada em *Web Services*.

Este projeto é composto por dois servidores principais: o servidor *FrontEnd* e o servidor *BackEnd*, que comunicam entre si via RMI. Este projeto também permite a interação através de clientes baseados em *Web Services* SOAP (*Simple Object Access Protocol*) e REST (*Representational State Transfer*) que foram implementados separadamente.

2. Arquitetura da Solução

2.1 Estrutura Geral

Este sistema segue uma arquitetura cliente-servidor distribuída, dividida entre plataformas Windows e Linux. Abaixo estão descritos os principais componentes do mesmo:

Servidor FrontEnd:

- Tecnologias Utilizadas:
 - Servidor Apache Tomcat 9
 - JAX-RS (para REST) e JAX-WS (para SOAP)
- Responsabilidades:
 - Gerir a interação com os utilizadores, incluindo registo, autenticação e encaminhamento de pedidos ao BackEnd.
 - Disponibilizar dois tipos de clientes, implementados separadamente:
 - Cliente REST: Utiliza HTTP e comunicação em formato JSON.
 - Cliente SOAP: Utiliza mensagens XML para comunicação SOAP (WSDL).
 - Armazenar os dados de utilizadores num ficheiro "utilizadores.txt".

Servidor BackEnd:

- Tecnologias Utilizadas:
 - Java RMI
- Responsabilidades:
 - Armazenar e gerir todas as informações sobre reservas, como criação, cancelamento e listagem de consultas.
 - Interagir com o FrontEnd via RMI.
 - Manter os dados em ficheiros clinicas.txt e consultas.txt.

Base de Dados em Ficheiros:

- utilizadores.txt: Armazena os dados de autenticação dos utilizadores.
Exemplo de formato:
name1,user1@email.com,nrTel,nrUtente,password123
name2,user2@email.com,nrTel,nrUtente,password456

- `clinicas.txt`: Contém informações sobre as clínicas e respectivas especialidades.
Exemplo de formato:
`ClinicaA,Clinica Geral,Dr. A;Dr. B`
`ClinicaA,Pediatria,Dr. C;Dr. D`
`ClinicaB,Hematologia,Dr. E`
- `consultas.txt`: Regista as consultas agendadas pelos utilizadores.
Exemplo de formato:
`C1234,name1,11/12/2024 08:00ClinicaA,Clinica Geral,Dr. A`
`C1235,name2,12/12/2024 19:00,ClinicaB,Hematologia,Dr. E`

2.2 Fluxo de Dados

1. O utilizador interage com o cliente (REST ou SOAP).
2. O cliente envia os pedidos ao servidor FrontEnd.
3. O servidor FrontEnd processa os pedidos e comunica com o BackEnd via RMI para executar as operações necessárias.
4. O BackEnd retorna os resultados ao FrontEnd, que os apresenta ao cliente.

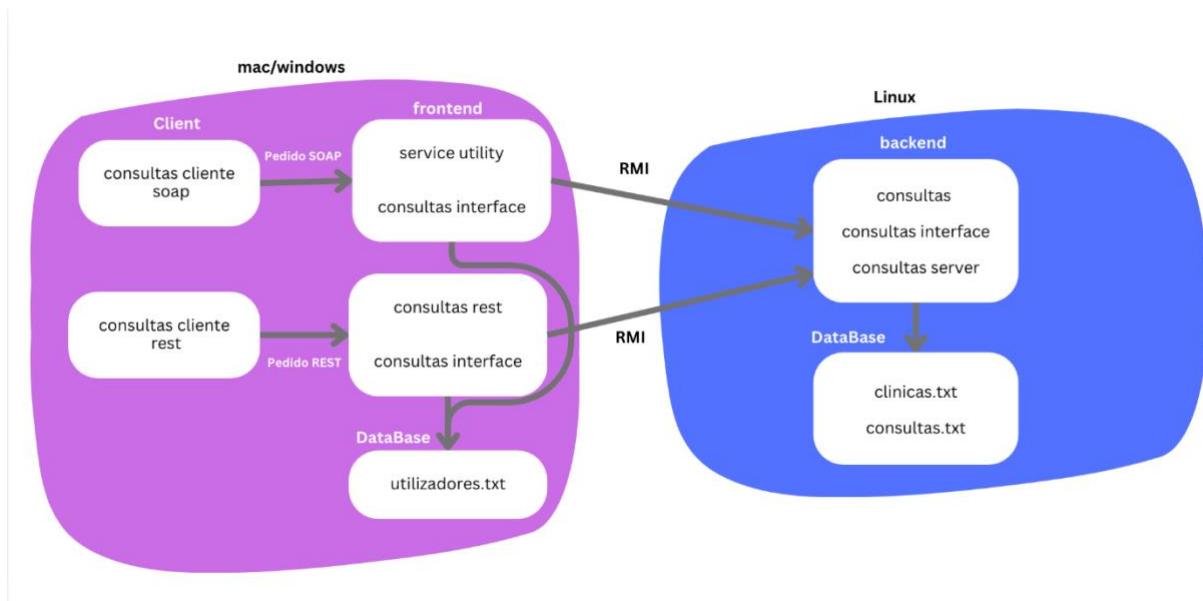
2.3 Funcionalidades Implementadas

O sistema disponibiliza as seguintes operações:

- Reservar uma consulta: Escolha da especialidade, clínica, data e hora.
- Cancelar uma consulta: Cancelamento de uma consulta existente através do ID.
- Listar consultas: Listagem de todas as consultas marcadas por um utilizador autenticado.
- Registar um utilizador: Criação de uma conta de utilizador.

2.4 Diagrama da Arquitetura

A Figura 1 ilustra a arquitetura do sistema, destacando os principais componentes, a divisão entre as plataformas Windows e Linux, e as interações entre os elementos:



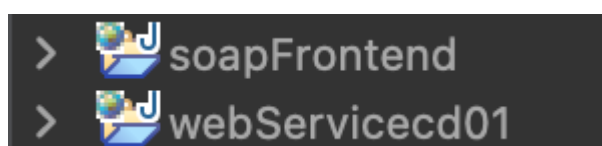
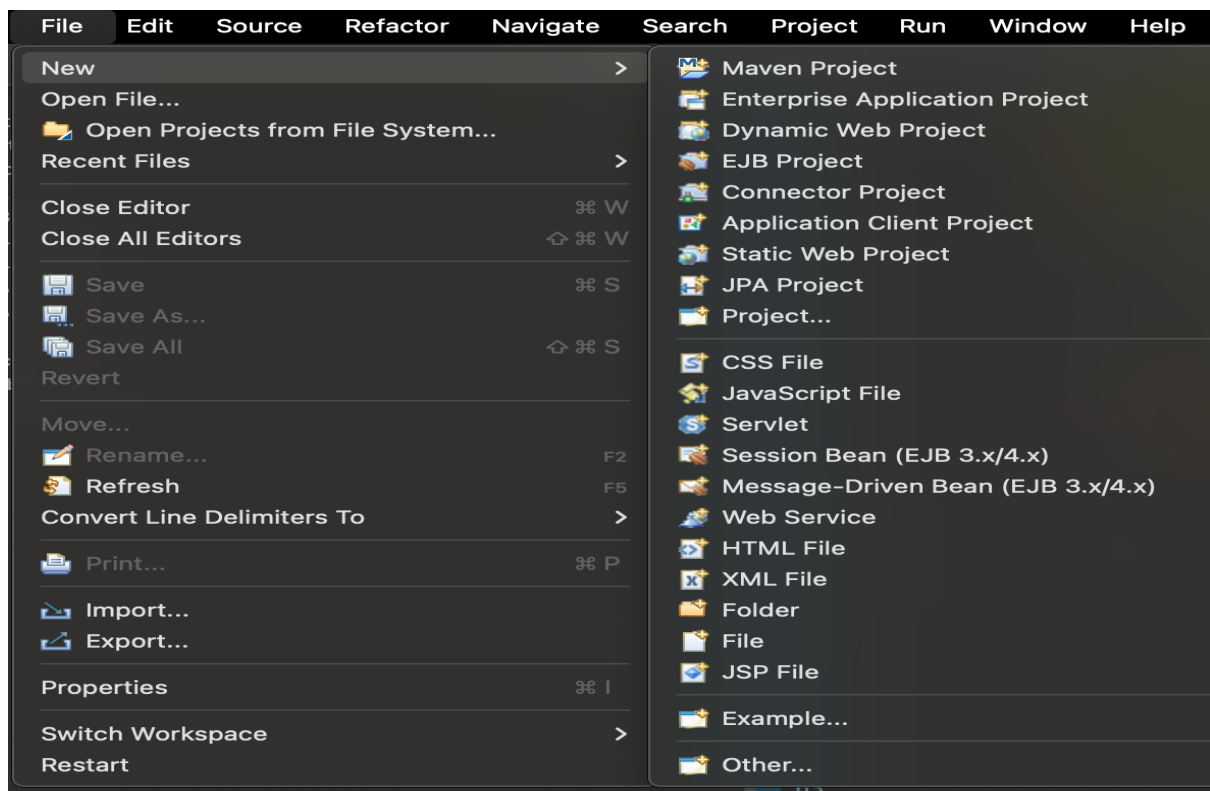
Legenda: Figura 1 - Arquitetura do sistema de marcação de consultas clínicas.

3. Manual de Instalação

Manual de Instalação do SOAP:

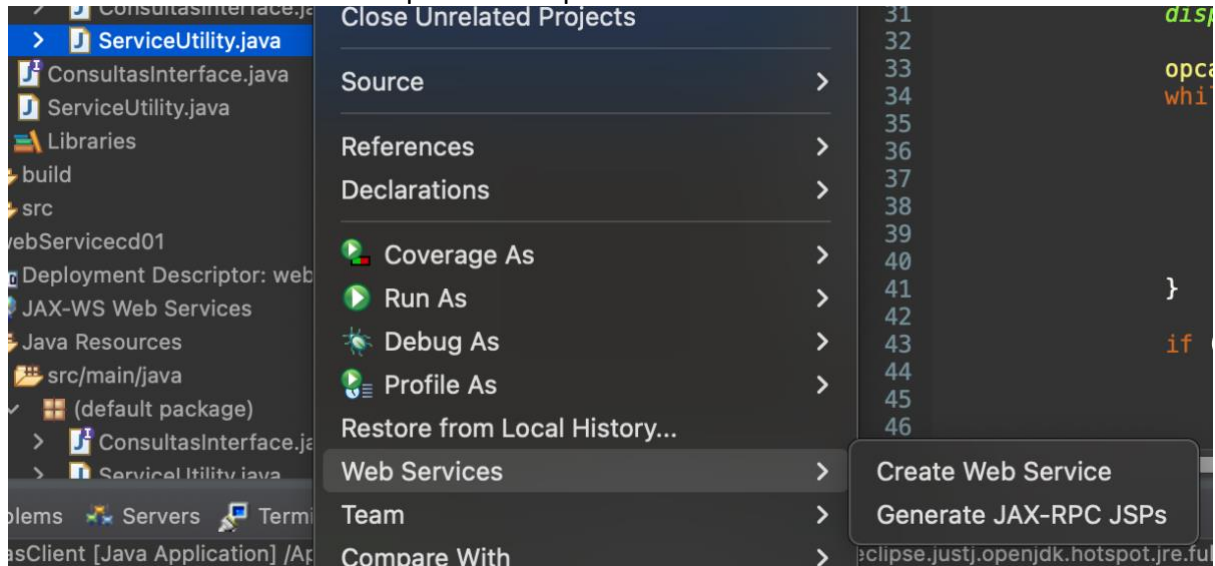
1. Criar os projetos

Crie dois *Dynamic Web Projects* no Eclipse: Um para o cliente SOAP. Outro para o *FrontEnd* SOAP.



2. Configuração do *FrontEnd* SOAP

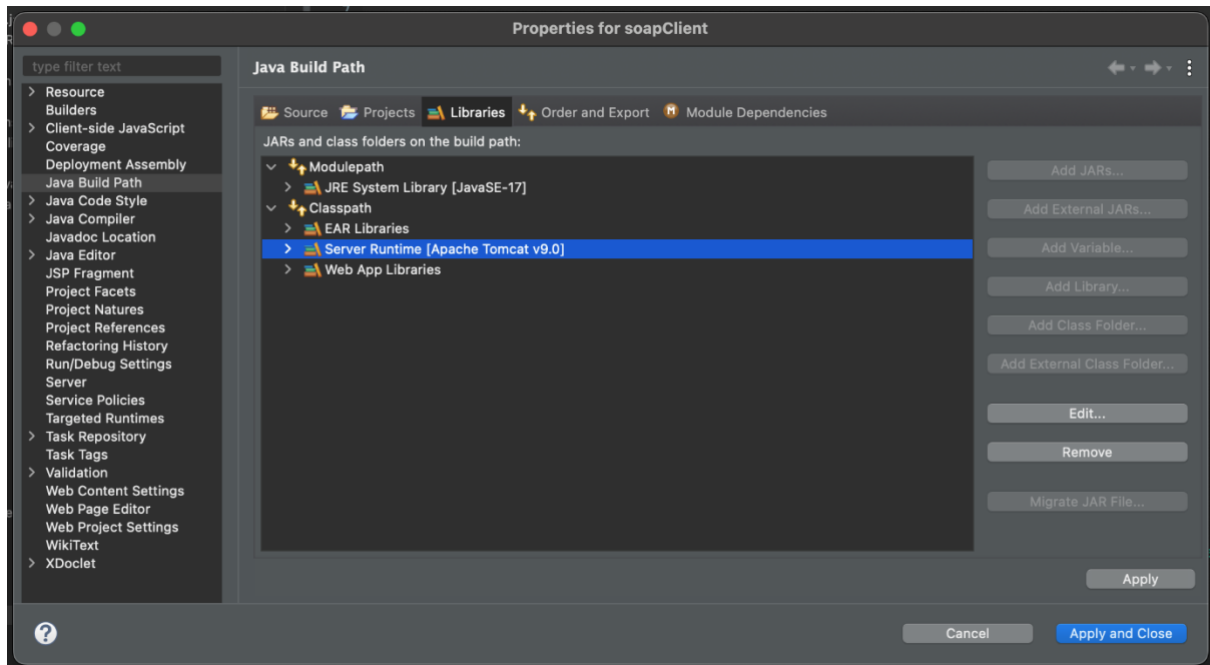
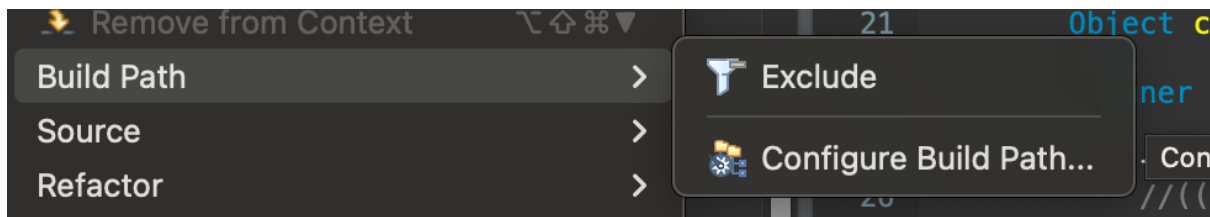
- No projeto do *FrontEnd*, copie os arquivos abaixo para `src/main/java`:
 - `ConsultasInterface.java`
 - `ServiceUtility.java`
- Configure o *Web Service*:
- Elimine o segundo import no arquivo “`cxf-beans.xml`”.
- Recrie o *Web Service* a partir do arquivo atualizado.



Available SOAP services:	
ServiceUtility <ul style="list-style-type: none">• listaHorario• listaClinicas• listaMedico• autenticarUtilizador• cancelarConsulta• registarUtilizador• seleccionarEspecialidade• reservarConsulta• seleccionarHorario• seleccionarMedico• listaEspecialidades• listarConsultas• seleccionarClinica	Endpoint address: <code>http://localhost:8080/soapFrontend/services/ServiceUtilityPort</code> WSDL : http://default_package/ServiceUtilityService Target namespace: <code>http://default_package/</code>

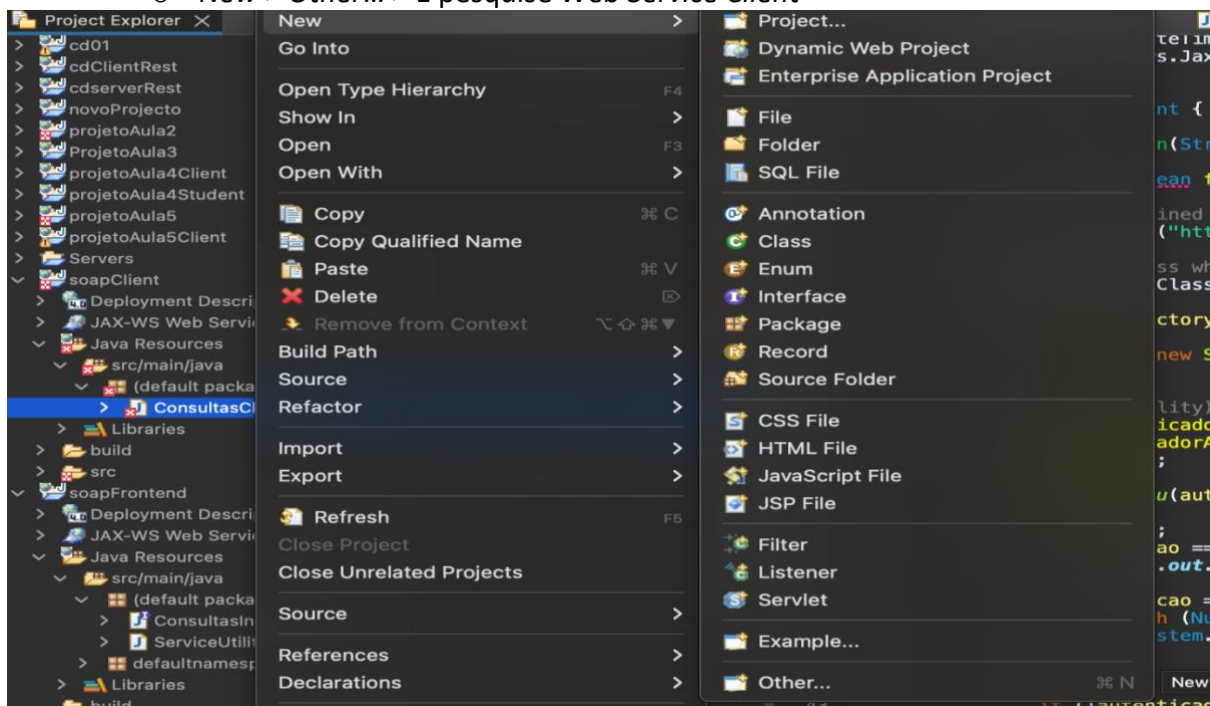
3. Configuração do cliente SOAP

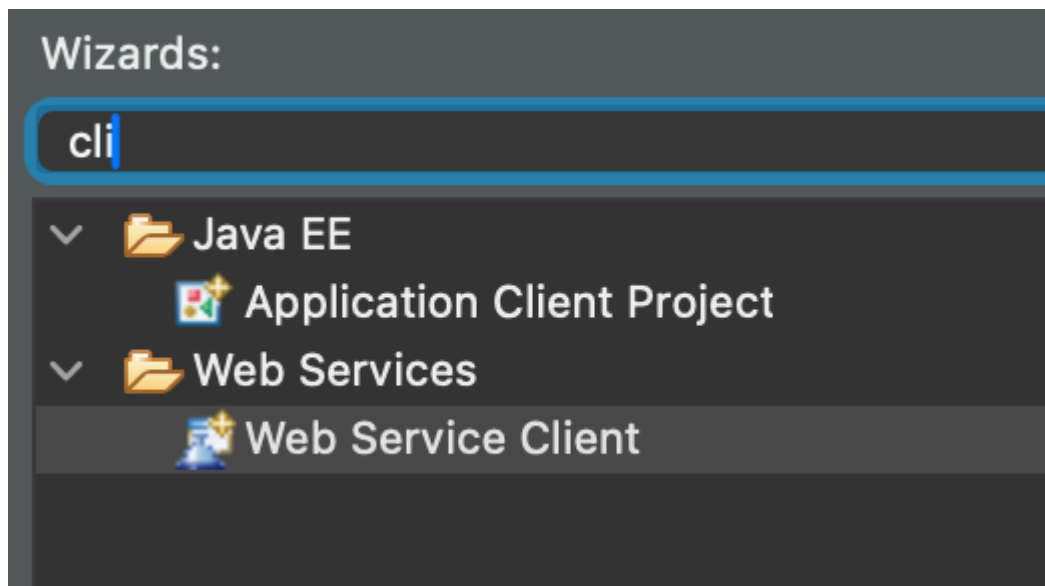
- No projeto do cliente, copie o arquivo `ConsultasClient.java` para o diretório `src/main/java`.
- Configure o cliente SOAP:
 - Insira o seguinte link no campo WSDL:
 - `http://localhost:8080/soapFrontend/services/ServiceUtilityPort?wsdl`.



4. Criar o Web Service Client

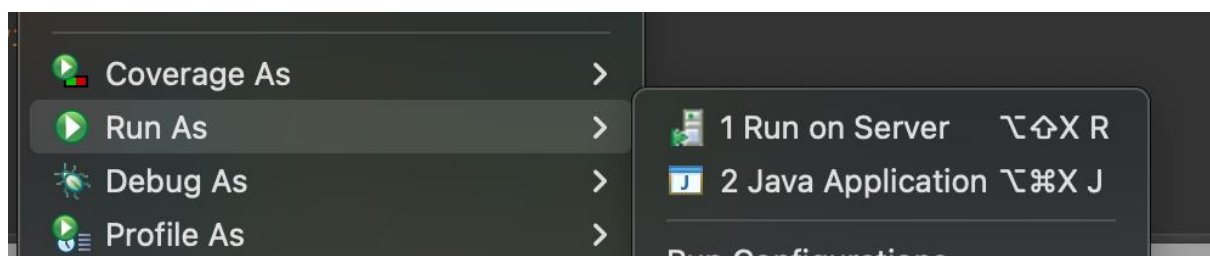
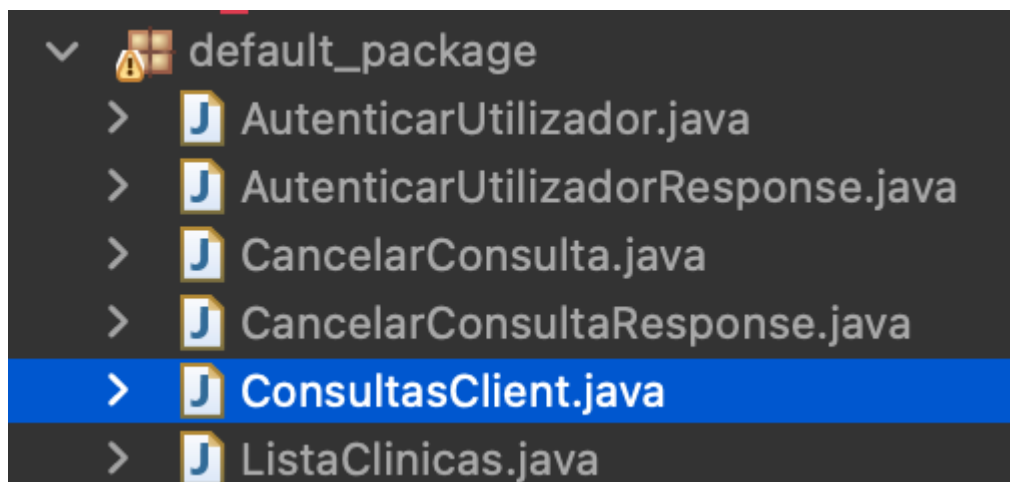
- Criar o *Web Service Client* na classe ConsultasClient:
 - Clique com o botão direito na classe ConsultasClient
 - *New > Other... > E* pesquise *Web Service Client*





5. Ajustes finais

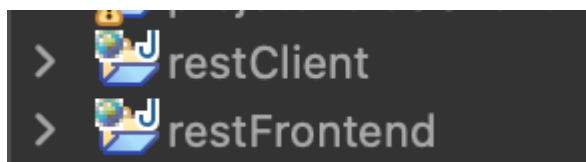
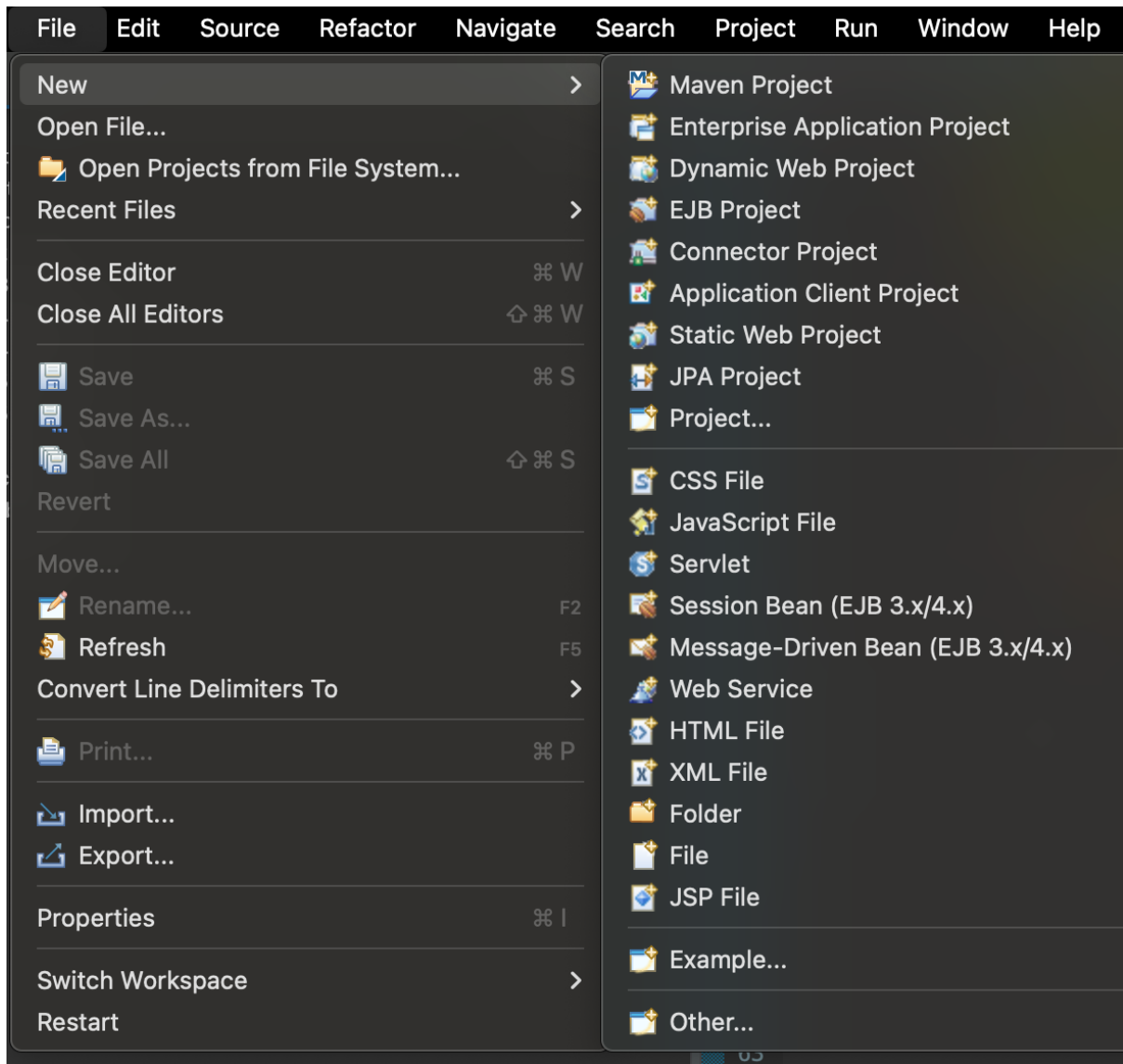
- No projeto do cliente SOAP:
 - Mova o arquivo ConsultasClient.java para o default package.
 - Execute o arquivo ConsultasClient.java como uma aplicação Java (*Run as > Java Application*).



Manual de Instalação do REST:

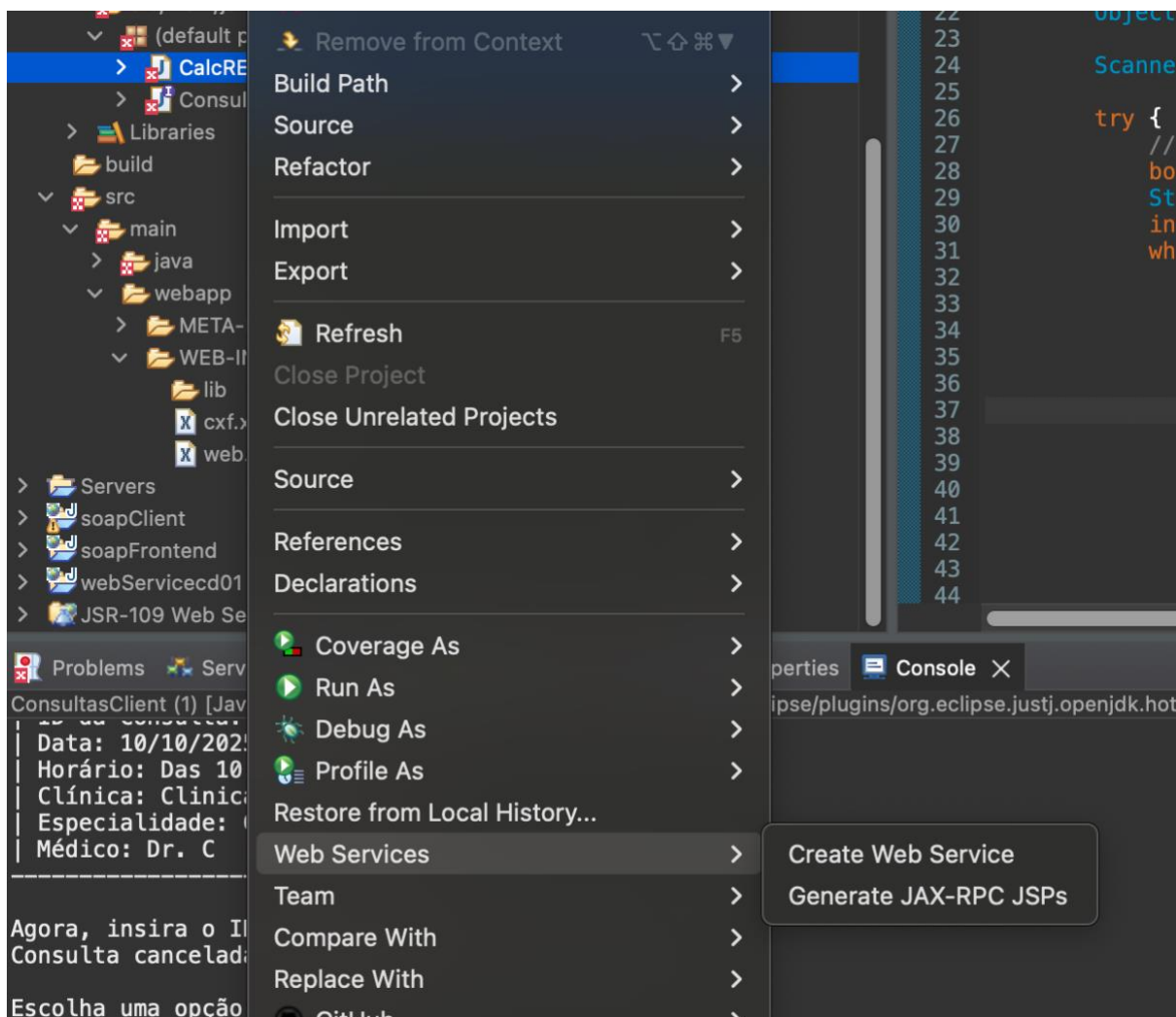
1. Criar os projetos

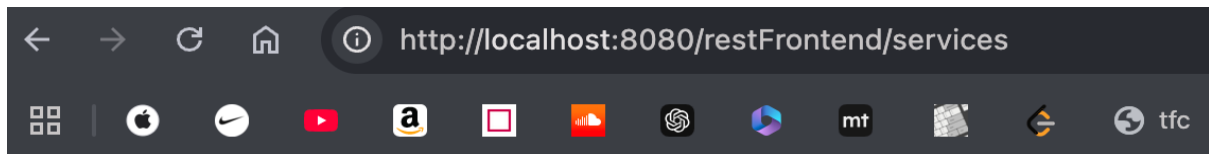
- Crie dois *Dynamic Web Projects* no Eclipse:
 - Um para o cliente REST.
 - Outro para o *FrontEnd* REST.



2. Configuração do *FrontEnd* REST

- No projeto do *FrontEnd*:
 - Copie os arquivos CalcRest.java e ConsultasInterface.java para o diretório src/main/java.
 - Copie os arquivos cxf.xml e web.xml para a pasta WEB-INF.
- Configure o *Web Service*:
 - Abra o arquivo CalcRest.java.
 - Crie o *Web Service* REST a partir do código.



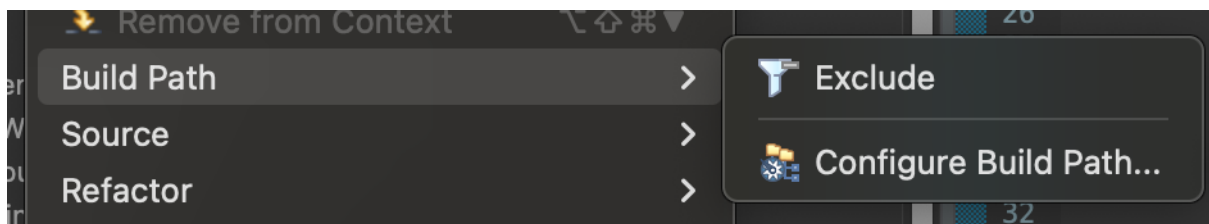
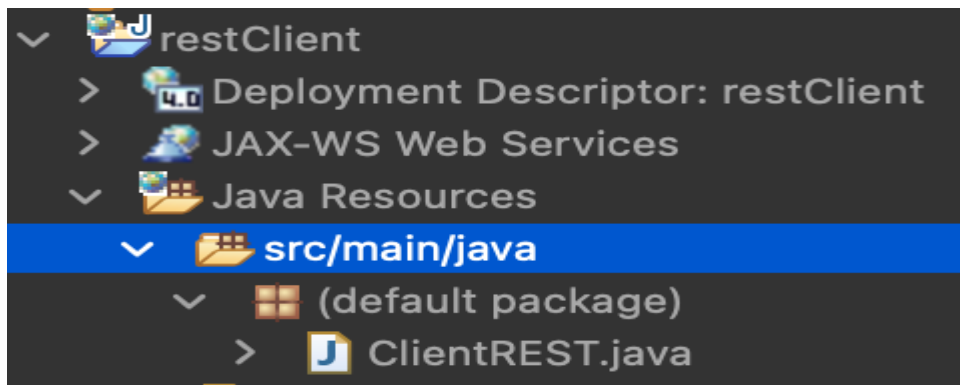


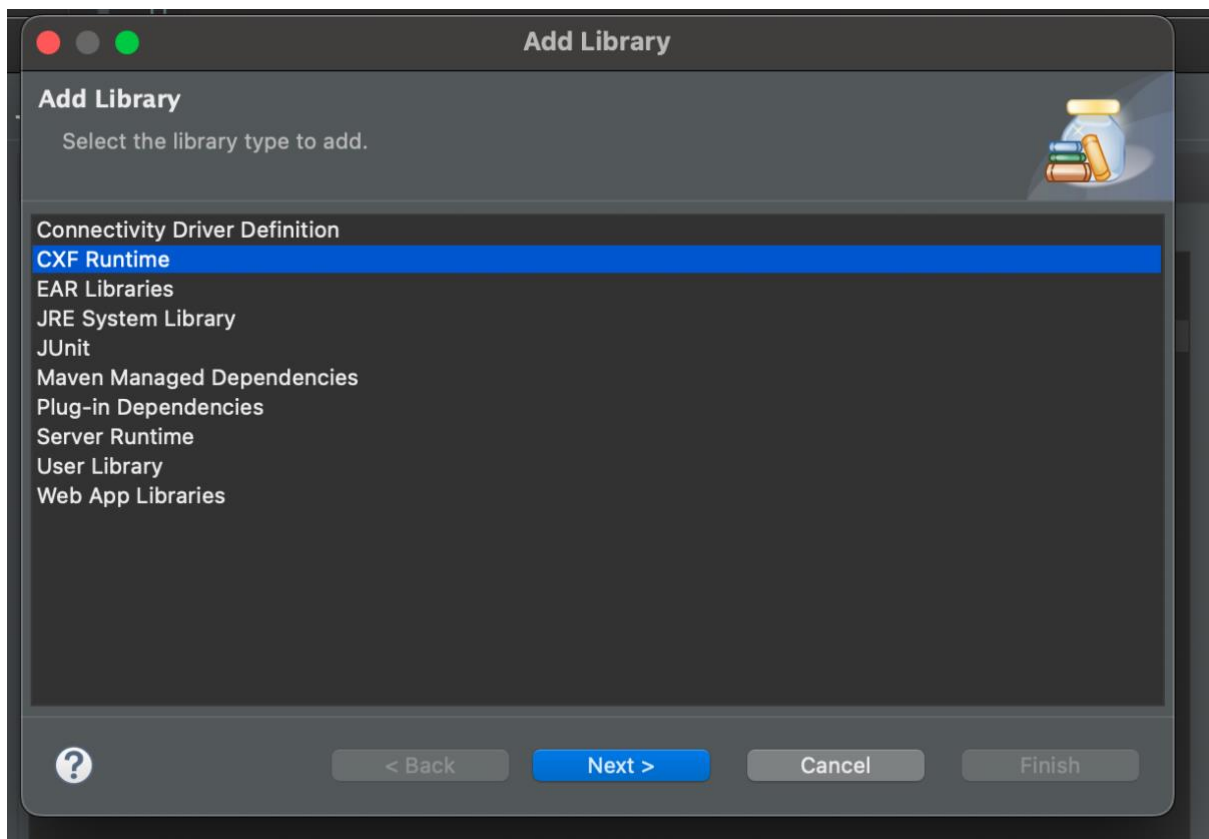
Available RESTful services:

Endpoint address: <http://localhost:8080/restFrontend/services/rest>
WADL : <http://localhost:8080/restFrontend/services/rest? wadl>

3. Configuração do cliente REST

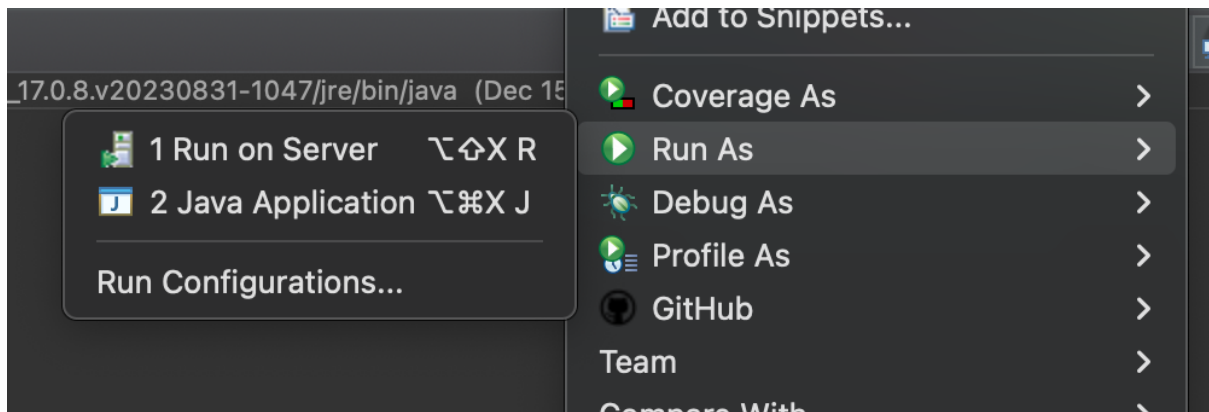
- No projeto do cliente REST:
 - Copie o arquivo ClientRest.java (ou similar) para o projeto.
- Configure o *classpath*:
 - Clique com o botão direito no projeto REST *Client*.
 - Navegue para *Build Path* > *Configure Build Path*.
 - Adicione o *CXF Runtime* ao *classpath* do projeto.





4. Executar o cliente REST

- Execute o arquivo principal do cliente REST como uma aplicação Java (*Run as > Java Application*).



Manual de Instalação do *BackEnd* RMI

1. Introdução

Este manual fornece as instruções necessárias para a instalação e execução do servidor RMI do *BackEnd* do sistema de marcação de consultas clínicas.

2. Estrutura dos Ficheiros

Os ficheiros necessários estão contidos na pasta rmi-example e incluem:

- `clinicas.txt` e `consultas.txt`: Arquivos que servem como a base de dados.
- `ConsultasServer.java`: Implementação do servidor RMI.
- `ConsultasInterface.java`: Interface que define os métodos RMI.
- `Consultas.java`: Implementação da lógica do servidor (métodos definidos na interface).

A estrutura de diretórios é a seguinte:

Transferências/Projeto/

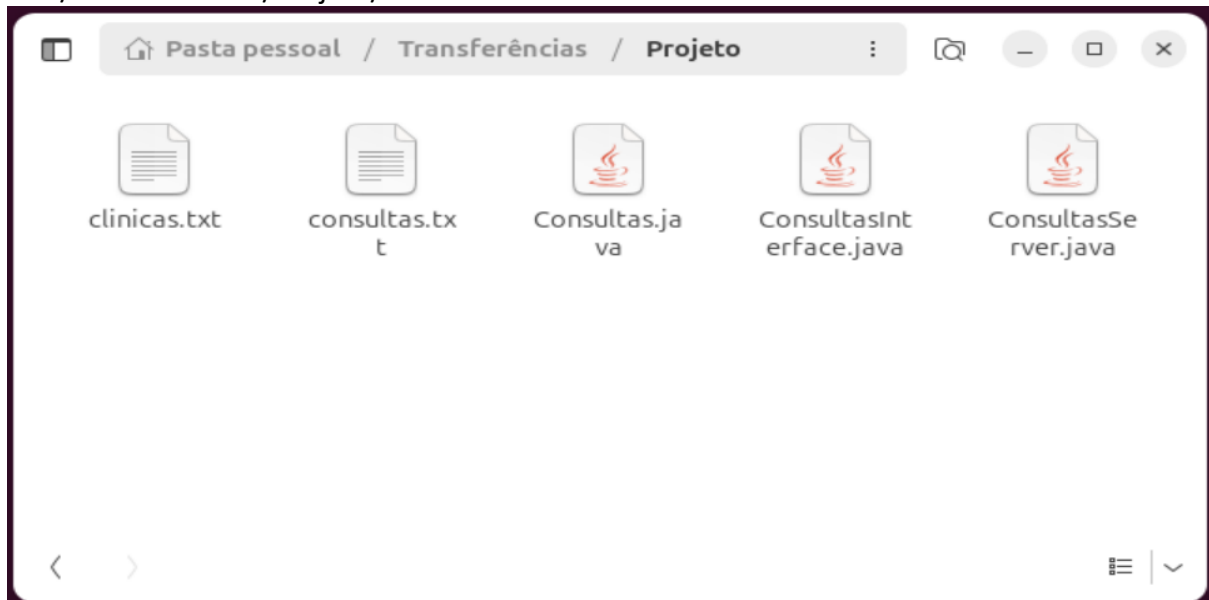
```
|-- clinicas.txt
|-- consultas.txt
|-- Consultas.java
|-- ConsultasInterface.java
|-- ConsultasServer.java
```

3. Compilação e Execução

Passo 1: Aceda ao Diretório Correto

Abra o terminal e navegue até à pasta onde os ficheiros estão localizados:

`cd ~/Transferências/Projeto/`



```
daniel@daniel-VirtualBox: ~/Transferências/Projeto
daniel@daniel-VirtualBox:~/Secretária$ cd
daniel@daniel-VirtualBox:~$ cd ~/Transferências/Projeto/
daniel@daniel-VirtualBox:~/Transferências/Projeto$
```

Passo 2: Compile os Ficheiros Java

Para compilar todos os ficheiros .java, execute o comando:

```
javac *.java
```

Se a compilação for bem-sucedida, os arquivos .class serão gerados no mesmo diretório.

```
daniel@daniel-VirtualBox: ~/Transferências/Projeto
daniel@daniel-VirtualBox:~/Transferências/Projeto$ javac *.java
daniel@daniel-VirtualBox:~/Transferências/Projeto$
```

Passo 3: Descubra o Endereço IP

Execute o comando abaixo para descobrir o endereço IP da máquina:

```
ip addr
```

Anote o IP da interface correta (por exemplo, 192.168.1.X). Este IP será usado para executar o servidor RMI.

```
daniel@daniel-VirtualBox: ~/Transferências/Projeto
daniel@daniel-VirtualBox:~/Transferências/Projeto$ ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:a4:5a:30 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.110/24 brd 192.168.1.255 scope global dynamic noprefixroute enp0s3
        valid_lft 3314sec preferred_lft 3314sec
daniel@daniel-VirtualBox:~/Transferências/Projeto$
```

Passo 4: Inicie o Servidor RMI

Execute o servidor usando o comando:

```
java ConsultasServer <IP> <porta>
```

Substitua <IP> pelo endereço IP obtido no passo anterior.

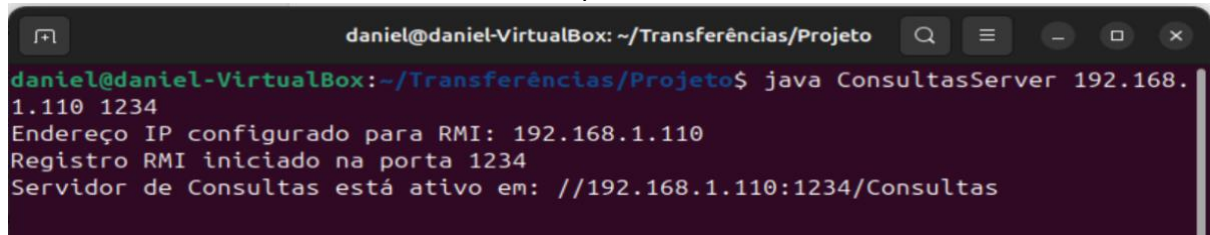
Substitua <porta> pela porta desejada para o servidor RMI (por exemplo, 1099).

Exemplo:

```
java ConsultasServer 192.168.1.110 1099
```

Se o servidor for iniciado corretamente, uma mensagem semelhante será exibida:

Servidor RMI iniciado no IP 192.168.1.110 e porta 1099.

A terminal window titled 'daniel@daniel-VirtualBox: ~/Transferências/Projeto' with standard window controls. The terminal shows the command 'java ConsultasServer 192.168.1.110 1234' being executed. The output consists of four lines: 'Endereço IP configurado para RMI: 192.168.1.110', 'Registro RMI iniciado na porta 1234', and 'Servidor de Consultas está ativo em: //192.168.1.110:1234/Consultas'.

```
daniel@daniel-VirtualBox: ~/Transferências/Projeto
daniel@daniel-VirtualBox:~/Transferências/Projeto$ java ConsultasServer 192.168.1.110 1234
Endereço IP configurado para RMI: 192.168.1.110
Registro RMI iniciado na porta 1234
Servidor de Consultas está ativo em: //192.168.1.110:1234/Consultas
```

4. Validação do Servidor

Para confirmar que o servidor está ativo:

Use um cliente RMI configurado para conectar ao servidor.

Teste a comunicação com as funcionalidades disponibilizadas (por exemplo, marcação de consultas).

5. Encerramento

Para encerrar o servidor, pressione Ctrl + C no terminal onde o servidor está a ser executado.

Manual de Utilização:

1. Exemplo de Autenticação:

```
Escolha uma opção:  
1. Registrar Utilizador  
2. Autenticar  
Opção: 2  
Email: ric@gmail.com  
Senha: lol  
Login bem sucedido! Bem-vindo, ric
```

2. Exemplo de Registo de Utilizador:

```
Escolha uma opção:  
1. Registrar Utilizador  
2. Autenticar  
Opção: 1  
Utilizador: danny  
Email: danny@gmail.com  
Numero de Telefone: 123456789  
Numero de Utente de saúde: 123456789  
Senha: danny123  
Sucesso: Utilizador registado com sucesso.  
  
Escolha uma opção:  
1. Registrar Utilizador  
2. Autenticar  
Opção: 2  
Email: danny@gmail.com  
Senha: danny123  
Login bem sucedido! Bem-vindo, danny
```

3. Exemplo de marcar uma consulta:

```
Escolha uma opção:
3. Reservar consulta
4. Cancelar consulta
5. Listar consultas
6. Sair
Opção: 3
Digite 'menu' a qualquer momento para voltar ao menu principal.
Selecione a clínica:
Clínicas disponíveis:
1. ClínicaA
2. ClínicaB
3. ClínicaC
4. ClínicaD
Número da clínica: 1
ClínicaA
Especialidades disponíveis na clínica ClínicaA:
1. Cardiologia
2. Clínica Geral
3. Ginecologia
4. Pediatria
Número da especialidade: 2
Médicos disponíveis para Clínica Geral:
1. Dr. Antonio Carvalho
2. Dr. Carlos Oliveira
3. Dr. Francisco Pereira
4. Dr. Joao Silva
5. Dr. Jose Santos
6. Dr. Miguel Costa
7. Dr. Paulo Almeida
8. Dr. Pedro Ferreira
9. Dr. Rui Martins
10. Dr. Tiago Rodrigues
Número do médico: 3
Digite a data (dd/mm/yyyy): 10/01/2025
Horários disponíveis:
08:00
09:00
10:00
11:00
12:00
13:00
14:00
15:00
16:00
17:00
18:00
19:00
Digite o horário disponível (hh:mm): 10:00
Consulta reservada com sucesso! ID: C8171
```

4. Exemplo de cancelamento de uma consulta:

```
Escolha uma opção:
3. Reservar consulta
4. Cancelar consulta
5. Listar consultas
6. Sair
Opção: 4
(Digite 'AJUDA' para listar as suas consultas)
ID da consulta para cancelamento: ajuda
=====
CONSULTAS
=====
-----
| ID da Consulta: C8171
| Data: 10/01/2025
| Horário: Das 10:00 as 11:00
| Clínica: ClínicaA
| Especialidade: Clínica Geral
| Médico: Dr. Francisco Pereira
-----
Agora, insira o ID da consulta para cancelamento: C8171
Consulta cancelada com sucesso!
```


5. Exemplo de visualização de uma consulta:

```
Escolha uma opção:
3. Reservar consulta
4. Cancelar consulta
5. Listar consultas
6. Sair
Opção: 5
=====
                        CONSULTAS
=====
-----
| ID da Consulta: C553
| Data: 10/01/2025
| Horário: Das 16:00 as 17:00
| Clínica: ClínicaB
| Especialidade: Clínica Geral
| Médico: Dr. Jose Pinto
-----
-----
| ID da Consulta: C1791
| Data: 11/02/2025
| Horário: Das 10:00 as 11:00
| Clínica: ClínicaA
| Especialidade: Clínica Geral
| Médico: Dr. Antonio Carvalho
-----
```

6. Exemplo de log out

```
Escolha uma opção:
3. Reservar consulta
4. Cancelar consulta
5. Listar consultas
6. Sair
Opção: 6
Sessão encerrada.
```