

Hotel de Charme

**Computação Distribuída**

Relatório e manual de configuração e utilização

João Pereira 21703345

Diogo Casaca 21604343

Sérgio Nunes

Computação Distribuída | LEI | 10/12/2021

Índice

[Índice 2](#_Toc91605803)

[Introdução 3](#_Toc91605803)

[Arquitetura 4](#_Toc91605803)

[Manual de instalação 7](#_Toc91605803)

[Manual de configuração 8](#_Toc91605803)

[Manual de utilização 21](#_Toc91605803)

**Introdução**

Este projeto consistiu no desenvolvimento de uma aplicação em que o objetivo foi a criação de um serviço de reserva de quartos no Hotel de Charme.

Esta aplicação foi dividida em Back-End e Front-End, sendo o back-end responsável pelo gerenciamento e armazenamento das reservas e o front-end pelo registo, autenticação e onde são disponibilizados os serviços implementados no back-end.

O hotel tem 3 pisos e é composto pelos seguintes quartos:

* Piso 1: 10 quartos para no máximo 2 pessoas, números (101 até 110);
* Piso 2: 10 quartos para no máximo 3 pessoas, números (201 até 210);
* Piso 3: 4 suites para no máximo 6 pessoas (301 até 304);

As funcionalidades disponíveis para os utilizadores são:

* Reservar quarto
* Cancelar quarto
* Listar quartos
* Registar utilizador
* Autenticar utilizador

**Arquitetura**

Diagram

Description automatically generated

Figura 1 - Arquitetura da solução

**Front-end**

O front-end foi implementado no sistema operativo Windows e é suportado por um servidor Apache Tomcat e a interação dos utilizadores com o front-end é realizado de duas formas distintas:

* **REST**

Através de uma aplicação cliente que emprega o protocolo HTTP e utiliza web services REST disponibilizados pelo servidor Tomcat.

As mensagens passadas através deste método são em formato JSON.

As operações HTTP usadas são GET e POST.

A operação GET é usada para pedir a lista de quartos disponíveis, passando a data inicial e final como parâmetros no url.

A operação POST é usada para o registo, autenticação, reserva de quarto e cancelamento de quarto, passando os respetivos parâmetros através do body.

* **SOAP**

Através de uma aplicação cliente que faz pedidos ao servidor e aguarda a sua reposta. A comunicação emprega mensagens XML SOAP e utiliza web services SOAP disponibilizados pelo servidor Tomcat.

O front-end é responsável pelo registo e autenticação dos utilizadores mas não tem qualquer informação sobre os quartos ou as reservas, essa informação reside unicamente no servidor back-end.

A comunicação entre o front-end e o back-end é feita através de RMI, uma interface de programação que permite a execução de chamadas remotas no estilo RPC.

**Back-end**

O back-end foi implementado no sistema operativo Linux, e é onde residem todas as funções de manipulação de dados dos quartos e reservas. Os dados das reservas são guardados em ficheiros onde são realizadas operações de leitura e escrita.

# Manual de instalação

Programas a instalar Front-End:

* Tomcat 9.0

<https://tomcat.apache.org/download-90.cgi>

* CXF Apache 3.4.5

<https://cxf.apache.org/download.html>

* Eclipse for Enterprise Java and Web Developer Edition

<https://www.eclipse.org/downloads/>

Programas a instalar Back-end

* VM Linux
* Servidor BackEnd

# Manual de configuração

**Back-end**

* Instalar a máquina virtual que contém o sistema operativo Linux

<https://drive.google.com/file/d/1urKfsEBdtrTt69Ui5H8xE0ajJcsFAnmW/view>

* + Palavra-chave de acesso á máquina virtual: “aluno-deisi”.
* Aceder através da linha de comandos á pasta que contém o projeto.
  + Exemplo Path: “aluno-deisi@linux-deisi-ulht:~/Downloads/CD\_Project21/src”
  + Caso seja necessário fazer download para a VM, aceder á pasta onde gravou o projeto.
* Compilar todos os ficheiros do projeto, introduzindo o seguinte comando na consola no *path* do projeto: javac \*.java
* Abrir um novo separador de consola no *path* do projeto e inicializar o rmi registry no port *default (port 1099)*
  + Introduzir o seguinte comando: rmiregistry
  + Caso o port 1099 já se encontre em uso, especificar o *port* a utilizar: rmiregistry 23123
* No primeiro separador de consola no *path* do projeto, vamos correr o servidor do *back end*, dependendo do *port*, introduzindo o seguinte comando.
  + Port Default: java -Djava.security.policy=policy AddServer
  + Port pré-definido: java -Djava.security.policy=policy AddServer 23123

Servidor estará pronto para receber pedidos dos seus clientes remotamente, poderá visualizar algumas notas na consola do servidor sobre os pedidos a serem recebidos pelo servidor.

# Front-end

* Após instalar as ferramentas necessárias em ambiente Windows/Linux/MacOs:
  + **Tomcat 9.0**
  + **CXF Apache 3.4.5**
  + **Eclipse for Enterprise Java and Web Developer Edition**
* No eclipse, configurar a ligação com o tomcat 9.0 e o CXF Apache 3.4.5 instalados anteriormente.
  + **Tomcat:** Aceder á tab ‘Window’ -> ‘Preferences’ -> ‘Server’ -> ‘Runtime Environments’, clicar em ‘ Add…’ e selecionar tomcat 9.0 e fazer Browse até ao tomcat instalado.

Graphical user interface, text, application

Description automatically generated

* **CXF Apache 3.4.5:** Aceder á tab ‘Window’ -> ‘Preferences’ -> ‘Web Services’ -> ‘CXF 2.x Preferences’ -> ‘CXF Runtime’ -> ‘ Add…’ -> ‘Browse..’ até á pasta onde gravou o CXF Apache, selecione o path -> Finish -> Apply and close.

Graphical user interface, text, application

Description automatically generated

**SOAP Web Service**

**Servidor SOAP**

* 1. No eclipse vamos abrir o ServerSoap disponibilizado. Clicar em ‘File’ -> ‘Open Projects from File System’.

Graphical user interface, text, application, email

Description automatically generated

Irá aparecer a seguinte janela, na qual seleciona o caminho para a pasta onde se encontra o ServerSoap clicando em ‘Directory’ e finalmente clica em ‘Finish’.

Graphical user interface, text, application, email

Description automatically generated

* 2. No projeto do ServerSoap, vamos criar um web service. Clicar em cima do ServerSoap -> ‘New’ -> ‘Other’ -> ‘Web Service’ -> ‘Service implementation’ -> selecionar a class ServerSoap -> Finish

Graphical user interface, text, application

Description automatically generated

Graphical user interface, application

Description automatically generated

* 3. Correr o servidor tomcat associado ao web service

Graphical user interface, application

Description automatically generated

* 4. Para verificar se o serviço foi publicado corretamente, aceder ao URL local: <http://localhost:8080/ServerSoap/services>.

Graphical user interface, text, application

Description automatically generated

´

**Cliente SOAP**

* 5. No eclipse vamos abrir o ClientSoap da mesma forma que abrimos o ServerSoap no ponto 1.
* 6. Repetir o ponto 2. Quando tiver na aba abaixo, em vez de selecionar Web Service, selecione Web Service Client

Graphical user interface, text, application

Description automatically generated

* 7. No url que utilizou no ponto 7. para verificar se o servidor já tinha disponibilizado o web service, acede ao link WSDL, e copiei o URL da página que foi redirecionado.

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

* 8. Copie o URL(<http://localhost:8080/ServerSoap/services/ServerSoapPort?wsdl>) para o Service definition da criação do Web Service Client no projeto ClientSoap.

Graphical user interface, text, application, email

Description automatically generated

* 9 Aceder ao ficheiro MathConsumer e selecionar ‘Run As’ -> ’Java Aplication’

Graphical user interface, text, application

Description automatically generated

* 10. Seguir as instruções da consola, para utilizar a aplicação.

**REST Web Service**

**Servidor Rest**

* 1. No eclipse vamos abrir o RestfullWebService disponibilizado. Clicar em ‘File’ -> ‘Open Projects from File System’.

Graphical user interface, text, application, email

Description automatically generated

Irá aparecer a seguinte janela, na qual seleciona o caminho para a pasta onde se encontra o RestfullWebService clicando em ‘Directory’ e finalmente clica em ‘Finish’.

Graphical user interface, text, application, email

Description automatically generated

* 2. No projeto do ServerSoap, vamos criar um web service. Clicar em cima do RestfullWebService -> ‘New’ -> ‘Other’ -> ‘Web Service’ -> ‘Service implementation’ -> selecionar a class RestConsumerImp -> Finish

Graphical user interface, application

Description automatically generatedGraphical user interface, text

Description automatically generated

* 3. Para correr o servidor, clique no RestfullWebService -> ‘Run As’ -> ‘1 Run on Server’.

Graphical user interface, application

Description automatically generated

Selecione o servidor Tomcat e clique em finish.

Graphical user interface, text, application

Description automatically generated

**REST Web Service**

**Servidor Rest**

* 4. No eclipse vamos abrir o RestfullWebServiceClient da mesma forma que abrimos o RestfullWebService no ponto 1.
* 5. Para correr o cliente navegue na pasta src/main/java até ao RestClient.java, no qual deve clicar -> ‘Run as’ -> ‘Java Application’.
* Graphical user interface, application

  Description automatically generated

# Manual de utilização

Quando inicia a aplicação, será lhe apresentado o seguinte menu de onde pode escolher se quer registar-se ou autenticar-se como demonstrado na seguinte imagem:

A picture containing logo

Description automatically generated

Para escolher uma opção basta escrever “1” se quiser registar-se ou “2” se quiser autenticar-se.

**Registo**

Se escolheu a opção registar, será lhe pedido para introduzir primeiro o seu email e seguidamente a sua password como demonstrado na seguinte imagem:

Graphical user interface, text

Description automatically generated

Caso o registo seja sucedido, é-lhe apresentado as seguintes frase:

Text

Description automatically generated with medium confidence

**Autenticação**

Após ter registado a sua conta terá de se autenticar com o respetivo email e password que registou.

Graphical user interface, text, application

Description automatically generated

Em caso de sucesso obterá a seguinte resposta:

Text

Description automatically generated

Caso o email ou a password não coincidam com os que usou no seu registo obterá as seguintes respostas respetivamente:

Logo

Description automatically generated with medium confidence

ou

A screenshot of a computer

Description automatically generated with medium confidence

Após se ter autenticado poderá escolher entre 3 possíveis ações:

Text

Description automatically generated

**Listar Quartos**

Para obter uma lista de quartos disponíveis terá primeiro de selecionar essa opção escrevendo “1”.

Após selecionada a opção, será lhe pedido o intervalo de datas no qual quer saber os quartos disponíveis.

Primeiro introduza a data início e seguidamente a data fim, como demonstra a imagem:

Graphical user interface, text, application

Description automatically generated

Obterá a resposta com todos os números dos quartos disponíveis entre as datas escolhidas.

A picture containing logo

Description automatically generated

**Reservar quarto**

Para reservar um quarto terá de escolher a opção “2”.

Apos selecionada a opção será lhe pedido os seguintes dados pela seguinte ordem: numero do quarto que pretende reservar, numero de hospedes, data inicio da reserva, data final da reserva e o seu email, como demonstra a imagem:

Text

Description automatically generated

Em caso de sucesso obterá a seguinte resposta:

Logo

Description automatically generated with medium confidence

Caso o quarto já se encontre reservado para as datas escolhidas obterá a seguinte resposta:

Text

Description automatically generated with low confidence

Caso escolha um número de hóspedes que exceda o limite obterá a seguinte resposta:

A picture containing logo

Description automatically generated

**Cancelar quarto**

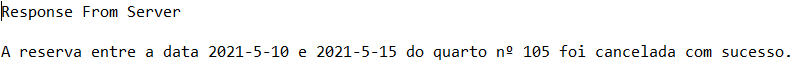
Para cancelar uma reserva de um quarto terá de escolher a opção “3”.

Após selecionada esta opção será lhe pedido para introduzir o numero do quarto da reserva que pretende cancelar, seguido das datas de inicio e fim da reserva.

Text

Description automatically generated

Caso o cancelamento seja bem-sucedido, será lhe apresentado a seguinte resposta:



Caso tentar cancelar uma reserva que não existe, será lhe apresentado o seguinte:

A picture containing text

Description automatically generated