



# DETI

## ***Back Engine***

*Instruções para instalação (RMI)*

António Rui Borges

# *Sumário*

- *Caracterização do problema*
- *Decomposição da solução*
- *Organização do pacote BackEngine*
  - *Descrição da estrutura hierárquica de directórios*
  - *Seu significado*
- *Construção do ambiente de execução*
  - *Compilação das diferentes packages*
  - *Seu deployment*
- *Lançamento da aplicação*
  - *Teste de acesso do servidor http ao directório público local*
  - *Componentes necessários*
  - *Configuração em plataforma única*
  - *Configuração multi-plataforma*

## *Caracterização do problema - 1*

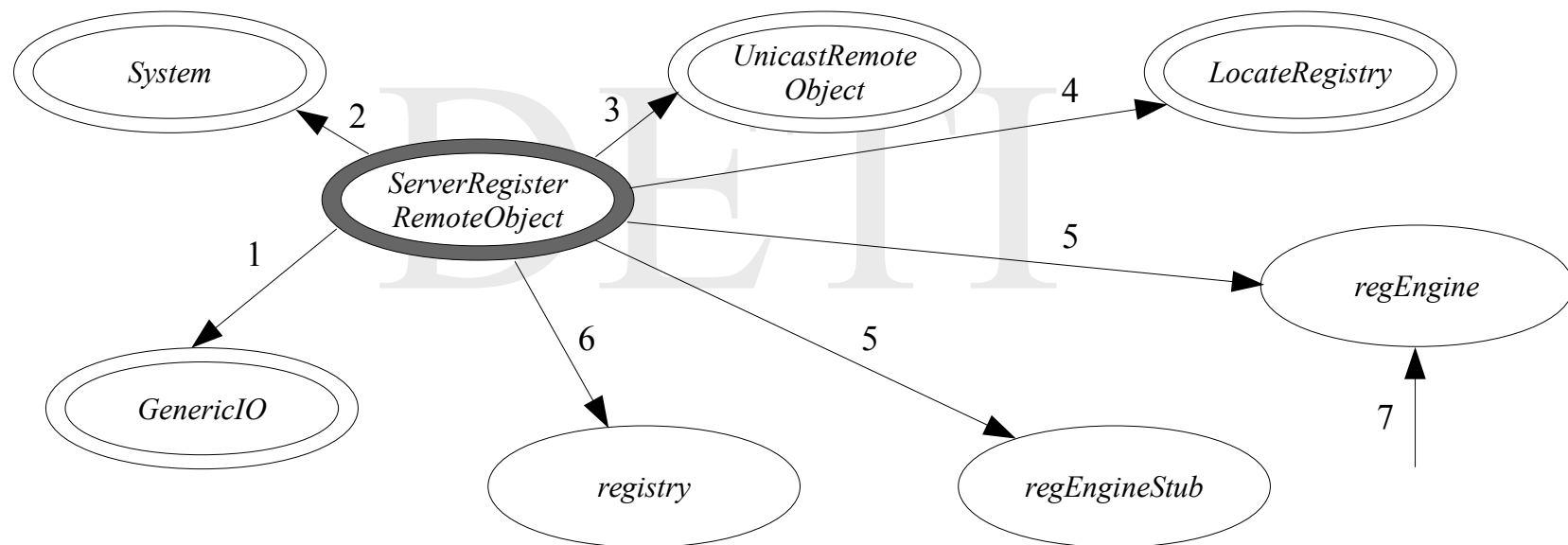
- trata-se de um exemplo, adaptado do tutorial da Sun sobre *RMI*, que pretende ilustrar o mecanismo de transferência de código entre máquinas virtuais de Java e sua execução remota;
- o problema consiste em duas entidades principais
  - um *objecto servidor*, de tipo `ComputeEngine`, que disponibiliza um serviço de execução local de código transferido para execução sob controlo remoto;
  - distintos *objectos clientes* que transferem o código a ser executado remotamente sob controlo local (interfaces `Compute` e `Task`); no caso presente, existe um objecto, de tipo `ComputePi`, que transfere um objecto de tipo `Pi` (cálculo de  $\pi$  com um número variável de algarismos decimais) para execução remota;

## *Caracterização do problema - 2*

- e três entidades auxiliares
  - um serviço de registo de objectos para acesso remoto (`rmiregistry`);
  - um *objecto servidor*, de tipo `RegisterRemoteObject`, que disponibiliza um serviço apoio ao registo de objectos remotos (residentes em máquinas virtuais de Java localizadas em plataformas hardware diferentes daquela que inclui o serviço de registo);
  - um *servidor http* que possibilita o *downloading* dinâmico de tipos de dados usados nas invocações remotas.

## Decomposição da solução - 1

### Lado do servidor de apoio ao registo de objectos remotos

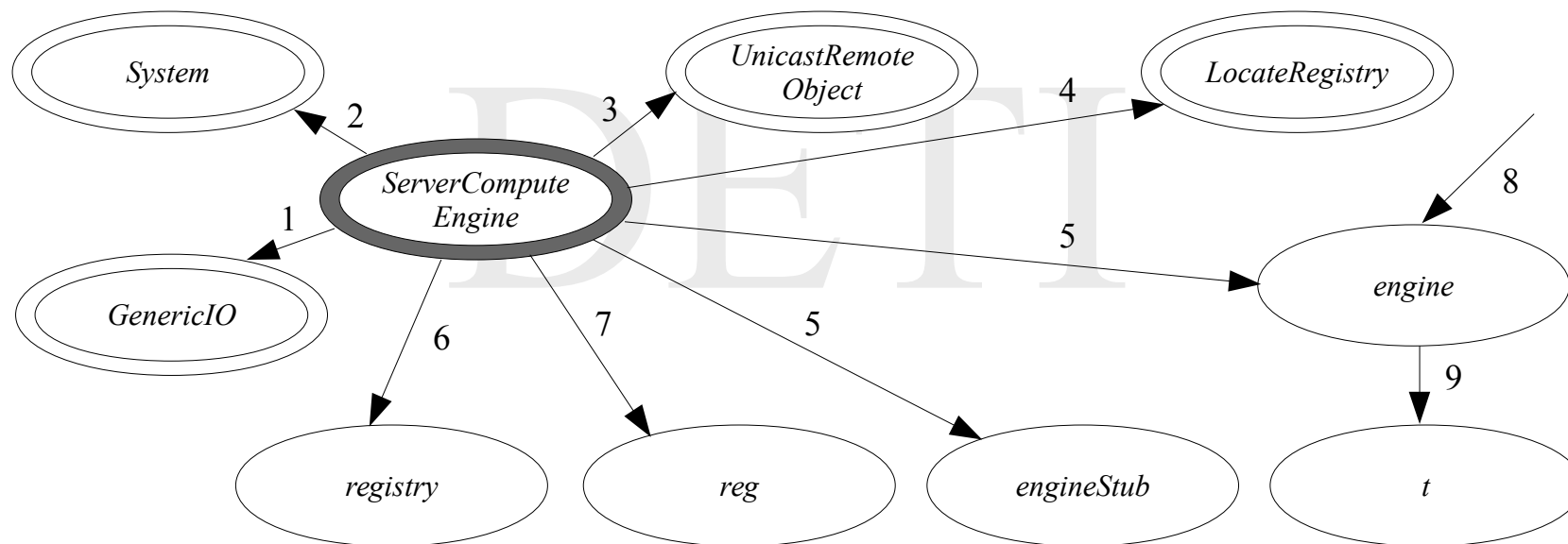


1 – readInt, readString, writeString, writeIntString  
2 – getSecurityManager, setSecurityManager  
3 – exportObject  
4 – getRegistry

5 – instantiate  
6 – instantiate, rebind  
7 – bind, unbind, rebind

## Decomposição da solução - 2

### *Lado do servidor de execução local de código móbil sob controlo remoto*

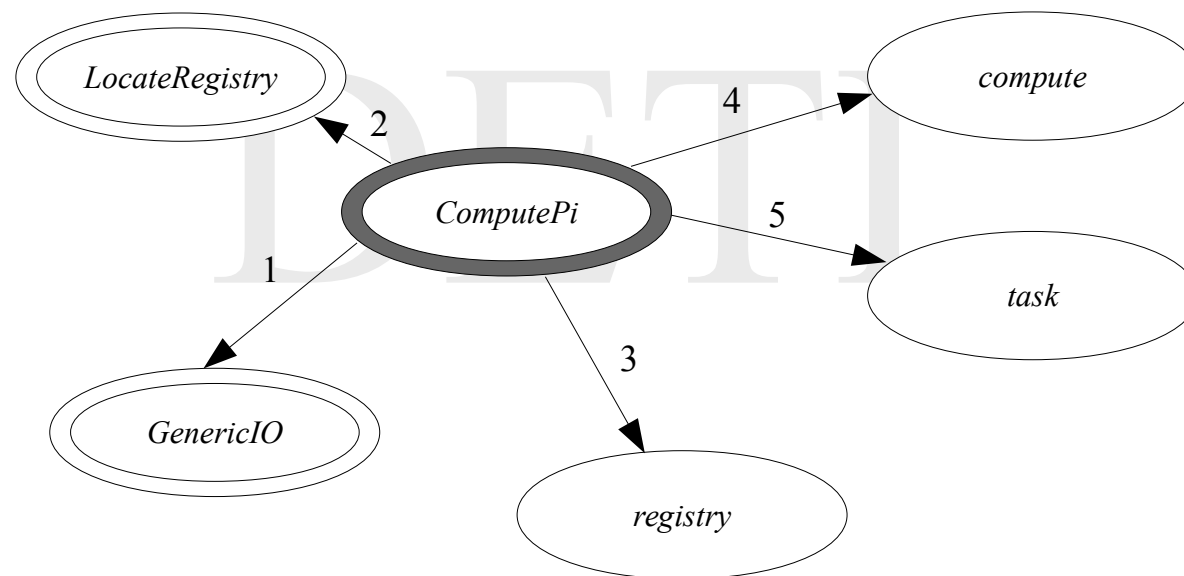


1 – readInt, readString, writeString, writeInt  
2 – getSecurityManager, setSecurityManager  
3 – exportObject  
4 – getRegistry  
5 – instantiate

6 – instantiate, locate  
7 – instantiate, bind  
8 – executeTask  
9 – execute

## *Decomposição da solução - 3*

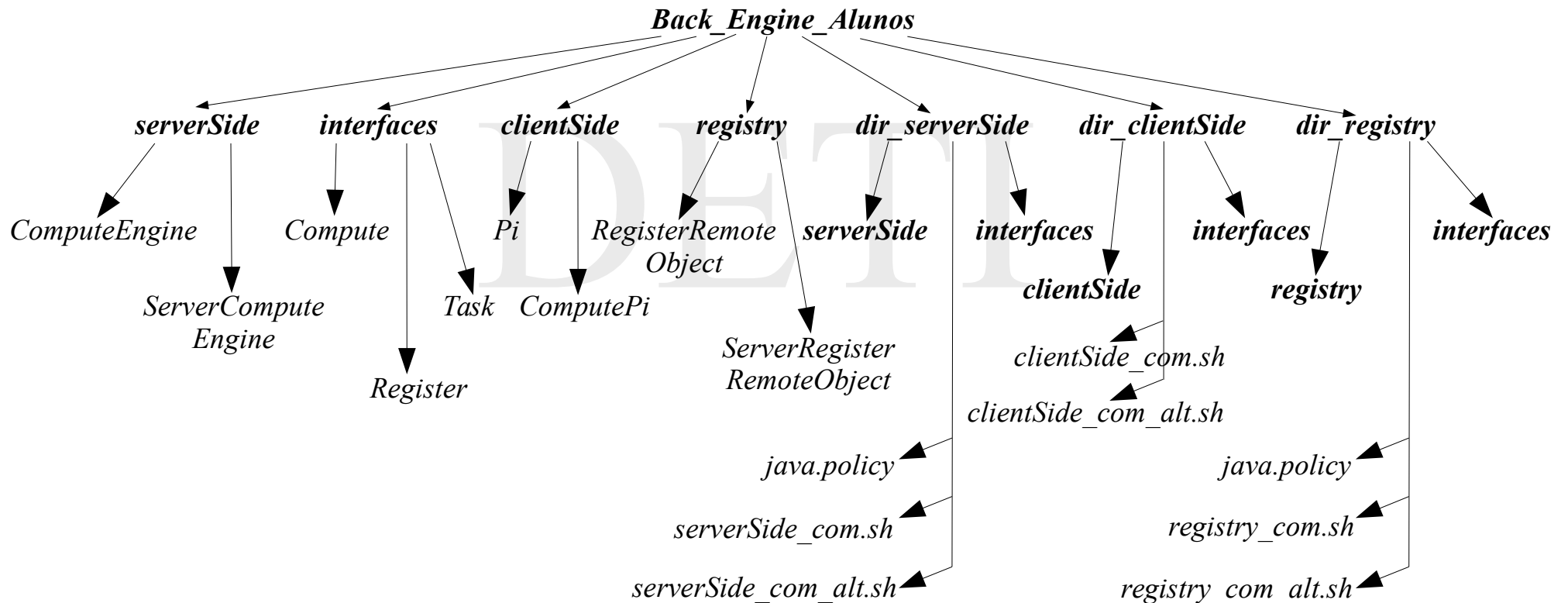
### *Lado do cliente*



1 – readInt, readString, writeString, writeString  
2 – getRegistry  
3 – instantiate, lookup

4 – instantiate, executeTask  
5 – instantiate

## Organização do pacote *BackEngine* - 1





## *Organização do pacote BackEngine - 2*

### *Região de desenvolvimento da aplicação*

*serverSide* – directório com o código do lado do servidor

*ComputeEngine*[.java] – código do serviço de execução local de código transferido para execução sob controlo remoto

*ServerComputeEngine*[.java] – código de instanciação e registo do objecto que providencia o serviço

*interfaces* – directório com o código dos interfaces de objectos para acesso remoto

*Compute*[.java] – interface de acesso ao objecto que providencia o serviço de execução remota sob controlo local de código que é transferido

*Task*[.java] – interface para execução do código móbil

*Register*[.java] – interface de acesso ao objecto que providencia o serviço de apoio ao registo de objectos remotos

*clientSide* – directório com o código do lado do cliente

*Pi*[.java] – código móbil que vai ser executado remotamente sob controlo local

*ComputePi*[.java] – código de acesso ao objecto servidor remoto para execução de métodos sobre ele

*registry* – directório com o código do servidor de apoio ao registo de objectos remotos

*RegisterRemoteObject*[.java] – código do serviço de apoio ao registo de objectos remotos

*ServerRegisterRemoteObject*[.java] – código de instanciação e registo do objecto que providencia o serviço

## *Organização do pacote BackEngine - 3*

### *Região de deployment da aplicação*

*dir\_serverSide* – directório para instalação do código do lado do servidor

*serverSide* – directório que irá conter: **ServerComputeEngine** [.class] e **ComputeEngine** [.class]

*interfaces* – directório que irá conter: **Register** [.class], **Compute** [.class] e **Task** [.class]

*java.policy* – ficheiro com a especificação da política de segurança local

*serverSide\_com.sh* – ficheiro de lançamento do servidor (configuração multi-plataforma)

*serverSide\_com\_alt.sh* – ficheiro de lançamento do servidor (configuração plataforma única)

*dir\_clientSide* – directório com o código do lado do cliente

*clientSide* – directório que irá conter: **ComputePi** [.class] e **Pi** [.class]

*interfaces* – directório que irá conter: **Compute** [.class] e **Task** [.class]

*clientSide\_com.sh* – ficheiro de lançamento do cliente (configuração multi-plataforma)

*clientSide\_com\_alt.sh* – ficheiro de lançamento do cliente (configuração plataforma única)

*dir\_registry* – directório com o código do servidor de apoio ao registo de objectos remotos

*registry* – directório que irá conter: **ServerRegisterRemoteObject** [.class] e **RegisterRemoteObject** [.class]

*interfaces* – directório que irá conter: **Register** [.class]

*java.policy* – ficheiro com a especificação da política de segurança local

*registry\_com.sh* – ficheiro de lançamento do servidor de apoio ao registo (configuração multi-plataforma)

*registry\_com\_alt.sh* – ficheiro de lançamento do servidor de apoio ao registo (configuração plataforma única)

## *Compilação das diferentes packages*

- criar uma janela de *shell*
- posicionar-se dentro do directório Back\_Engine\_Alunos
- compilar sucessivamente as *packages* que constituem a aplicação

```
[ruib@ruib-laptop Back_Engine_Alunos]$ pwd  
/home/ruib/Teaching/java/exemplos/Back_Engine_Alunos  
[ruib@ruib-laptop Back_Engine_Alunos]$ javac interfaces/*.java  
[ruib@ruib-laptop Back_Engine_Alunos]$ javac registry/*.java  
[ruib@ruib-laptop Back_Engine_Alunos]$ javac serverSide/*.java  
[ruib@ruib-laptop Back_Engine_Alunos]$ javac clientSide/*.java  
[ruib@ruib-laptop Back_Engine_Alunos]$
```

## *Deployment dos diferentes ficheiros executáveis*

- copiar os ficheiros \*.class para os diferentes directórios de execução

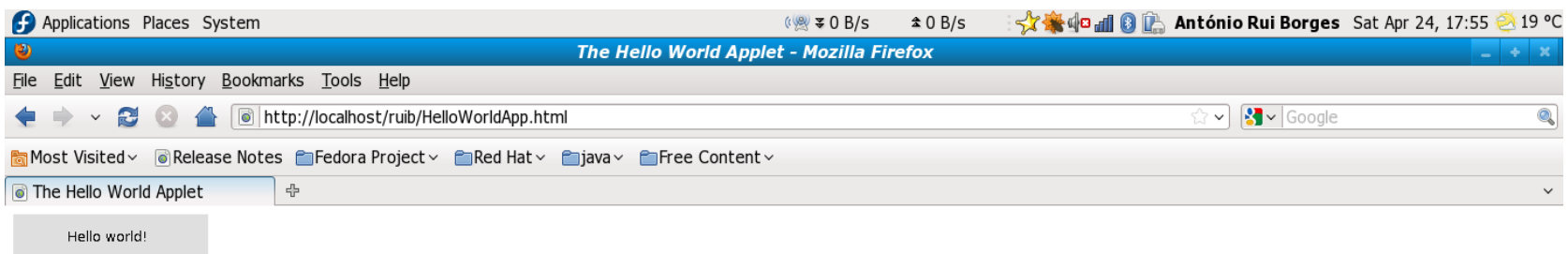
```
[... Back_Engine_Alunos]$ cp interfaces/Register.class dir_registry/interfaces/
[... Back_Engine_Alunos]$ cp registry/*.class dir_registry/registry/
[... Back_Engine_Alunos]$
[... Back_Engine_Alunos]$ cp interfaces/*.class dir_serverSide/interfaces/
[... Back_Engine_Alunos]$ cp serverSide/*.class dir_serverSide/serverSide/
[... Back_Engine_Alunos]$
[... Back_Engine_Alunos]$ cp interfaces/Compute.class interfaces/Task.class \
>dir_clientSide/interfaces/
[... Back_Engine_Alunos]$ cp clientSide/*.class dir_clientSide/clientSide/
[... Back_Engine_Alunos]$
```

## *Teste de acesso do servidor http ao directório público local - 1*

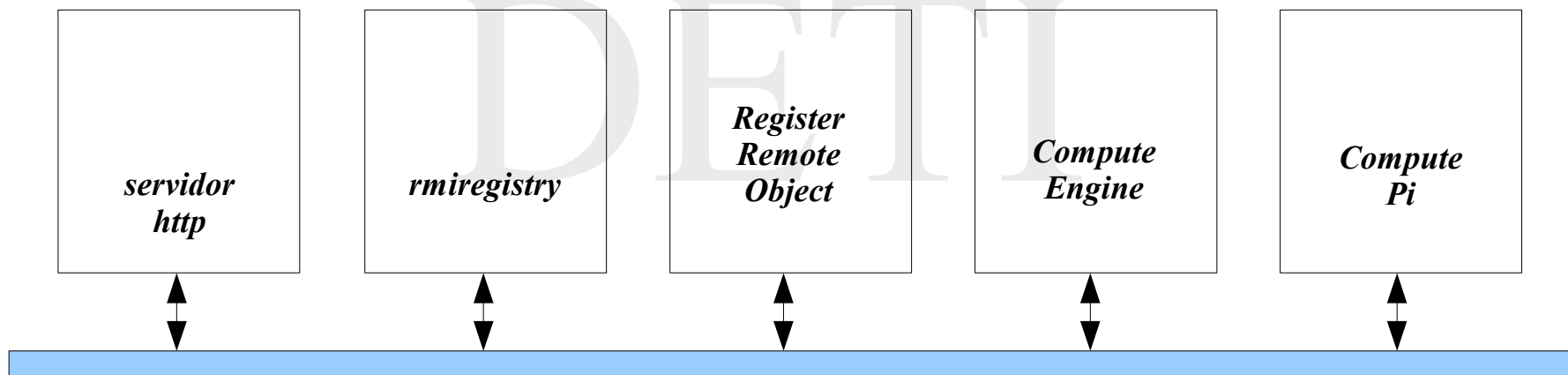
- garanta a existência de um directório de acesso público (permissões de leitura e execução para *todos*) localizado no directório base do utilizador
  - o nome tipicamente atribuído é `Public`, mas qualquer nome serve;
  - note que as permissões de leitura e execução para *todos* têm que ser mantidas desde a raiz da árvore de directórios;
- crie no directório base de acesso público do servidor *http* um *softlink* para o directório `Public`
  - a localização deste directório não é exactamente a mesma em todas as distribuições Linux, mas está normalmente presente em `/var` (`/var/www/html` para a Fedora);
- descarregue o pacote *Applet* do site de *e-learning* da disciplina, descomprima-o e armazene os ficheiros no directório `Public`;
- invoque uma janela de *browsing* e aceda ao site

*http://localhost/<nome do softlink>/HelloWorldApp.html*

## *Teste de acesso do servidor http ao directório público local - 2*



## *Componentes necessários - 1*



## *Componentes necessários - 2*

- o correcto funcionamento da aplicação pressupõe o lançamento sucessivo dos diferentes componentes
  - garantir que o servidor http está activo e operacional;
  - lançar o serviço de registo de objectos para acesso remoto (`rmiregistry`);
  - lançar o serviço apoio ao registo de objectos remotos residentes em máquinas virtuais de Java localizadas em plataformas hardware diferentes daquela que inclui o serviço de registo (`ServerRegisterRemoteObject`);
  - lançar o serviço de execução local de código transferido para execução sob controlo remoto (`ServerComputeEngine`);
  - lançar a aplicação cliente (`ComputePi`) que transfere um objecto de tipo `Pi` (cálculo de  $\pi$  com um número variável de algarismos decimais) para execução remota.



## Configuração em plataforma única - 1

- alteração do *shell script* de lançamento das máquinas virtuais de Java  
registry\_com\_alt.sh

```
[... dir_registry]$ pwd
```

```
/home/ruib/Teaching/java/exemplos/Back_Engine_Alunos/dir_registry
```

```
[... dir_registry]$ ls -l
```

```
total 20
```

```
drwxr-xr-x. 2 ruib ruib 4096 2010-04-24 11:04 interfaces
-rw-r--r--. 1 ruib ruib  196 2006-04-27 13:38 java.policy
drwxr-xr-x. 2 ruib ruib 4096 2010-04-24 11:06 registry
-rwxr-xr-x. 1 ruib ruib  184 2010-04-19 17:04 registry_com_alt.sh
-rwxr-xr-x. 1 ruib ruib  159 2010-04-19 17:05 registry_com.sh
```

```
[... dir_registry]$ cat registry_com_alt.sh
```

```
java -Djava.rmi.server.codebase="file:///home/ruib/Teaching/java/exemplos/Back_Engine/dir_registry/"\
-Djava.rmi.server.useCodebaseOnly=false\
-Djava.security.policy=java.policy\
registry.ServerRegisterRemoteObject
```

```
[... dir_registry]$
```

## Configuração em plataforma única - 2

- alteração do *shell script* de lançamento das máquinas virtuais de Java  
serverSide\_com\_alt.sh

```
[... dir_serverSide]$ pwd
```

```
/home/ruib/Teaching/java/exemplos/Back_Engine_Alunos/dir_serverSide
```

```
[... dir_serverSide]$ ls -l
```

```
total 20
```

```
drwxr-xr-x. 2 ruib ruib 4096 2010-04-24 11:40 interfaces
```

```
-rw-r--r--. 1 ruib ruib 196 2006-04-27 13:38 java.policy
```

```
drwxr-xr-x. 2 ruib ruib 4096 2010-04-24 11:48 serverSide
```

```
-rwxr-xr-x. 1 ruib ruib 181 2010-04-19 17:03 serverSide_com_alt.sh
```

```
-rwxr-xr-x. 1 ruib ruib 154 2010-04-19 17:04 serverSide_com.sh
```

```
[... dir_serverSide]$ cat serverSide_com_alt.sh
```

```
java -Djava.rmi.server.codebase="file:///home/ruib/Teaching/java/exemplos/Back_Engine/dir_serverSide/"\  
-Djava.rmi.server.useCodebaseOnly=false\  
-Djava.security.policy=java.policy\  
serverSide.ServerComputeEngine
```

```
[... dir_serverSide]$
```

## Configuração em plataforma única - 3

- alteração do *shell script* de lançamento das máquinas virtuais de Java  
clientSide\_com\_alt.sh

```
[... dir_clientSide]$ pwd
```

```
/home/ruib/Teaching/java/exemplos/Back_Engine_Alunos/dir_clientSide
```

```
[ruib@ruib-laptop dir_clientSide]$ ls -l
```

```
total 16
```

```
drwxr-xr-x. 2 ruib ruib 4096 2010-04-24 11:52 clientSide
```

```
-rwxr-xr-x. 1 ruib ruib 133 2010-04-20 08:19 clientSide_com_alt.sh
```

```
-rwxr-xr-x. 1 ruib ruib 29 2010-04-19 16:54 clientSide_com.sh
```

```
drwxr-xr-x. 2 ruib ruib 4096 2010-04-24 11:51 interfaces
```

```
[... dir_clientSide]$ cat clientSide_com_alt.sh
```

```
java -Djava.rmi.server.codebase="file:///home/ruib/Teaching/java/exemplos/Back_Engine/dir_clientSide/"\  
-Djava.rmi.server.useCodebaseOnly=false\  
clientSide.ComputePi $1
```

```
[... dir_clientSide]$
```

## *Configuração em plataforma única - 4*

- criação de quatro janelas de *shell* para lançamento dos diferentes componentes da aplicação

***Janela 1*** (garantir que o directório base não contém ficheiros \*.class)

```
[... Back_Engine_Alunos]$ cd  
[... Back_Engine_Alunos]$ pwd  
/home/ruib  
[... ~]$ set_rmiregistry_alt 22###
```

***Janela 2***

```
[... dir_registry]$ pwd  
/home/ruib/Teaching/java/exemplos/Back_Engine_Alunos/dir_registry  
[... dir_registry]$ registry_com_alt.sh  
Nome do nó de processamento onde está localizado o serviço de registo? localhost  
Número do port de escuta do serviço de registo? 22###  
Stub was generated!  
RMI registry was created!  
RegisterRemoteObject object was registered!
```

## Configuração em plataforma única - 5

### Janela 3

```
[... dir_serverSide]$ pwd
/home/ruib/Teaching/java/exemplos/Back_Engine_Alunos/dir_serverSide
[... dir_serverSide]$ serverSide_com_alt.sh
Nome do nó de processamento onde está localizado o serviço de registo? localhost
Número do port de escuta do serviço de registo? 22###
Stub was generated!
RMI registry was created!
ComputeEngine object was registered!
```

### Janela 4

```
[... dir_clientSide]$ pwd
/home/ruib/Teaching/java/exemplos/Back_Engine_Alunos/dir_clientSide
[... dir_clientSide]$ clientSide_com_alt.sh 10
Nome do nó de processamento onde está localizado o serviço de registo? localhost
Número do port de escuta do serviço de registo? 22###
3.1415926536
[... dir_clientSide]$
```

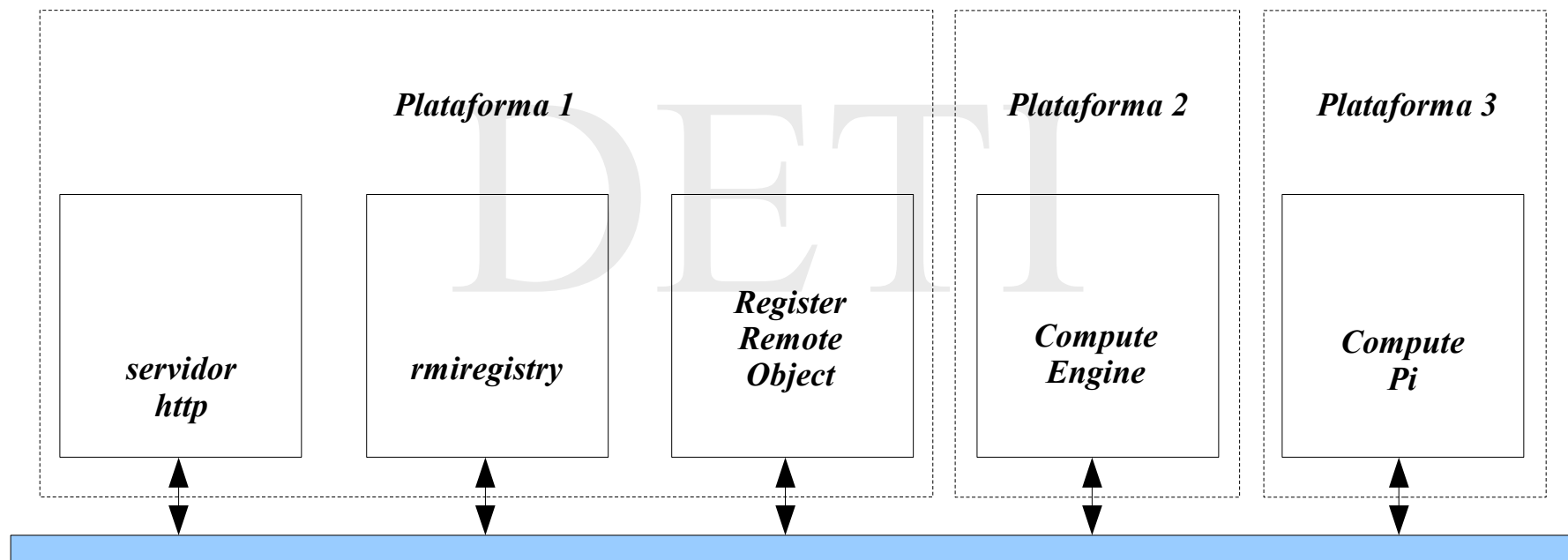
n.º de algarismos da parte decimal de  $\pi$

## Configuração multi-plataforma - 1

- preparação do directório `Public` para o *downloading* dinâmico de tipos de dados usados nas invocações remotas
  - crie no directório `Public` um directório de nome `classes`;
  - crie no directório `classes` dois directórios de nomes `interfaces` e `clientSide`, respectivamente;
  - copie os ficheiros `*.class`, associados com interfaces e com o código móbil, para os directórios de acesso;

```
[... Public]$ pwd
/home/ruib/Public
[... Public]$ mkdir classes
[... Public]$ mkdir classes/interfaces
[... Public]$ mkdir classes/clientSide
[... Public]$ cp ../Teaching/java/exemplos/Back_Engine_Alunos/interfaces/*.class \
>classes/interfaces
[... Public]$ cp ../Teaching/java/exemplos/Back_Engine_Alunos/clientSide/Pi.class \
>classes/clientSide
[... Public]$
```

## *Configuração multi-plataforma - 2*



## Configuração multi-plataforma - 3

- determinação do endereço IP do sistema computacional correspondente à plataforma 1
- isto é concretizado usando o comando *ifconfig*
  - *ifconfig eth0* (ligação por cabo);
  - *ifconfig wlan0* (ligação *wireless*);

```
[ruib@ruib-laptop ~]$ ifconfig wlan0
wlan0      Link encap:Ethernet  HWaddr 00:18:DE:A1:8C:FF
            inet addr:192.168.1.68  Bcast:192.168.1.255  Mask:255.255.255.0
            inet6 addr: fe80::218:deff:fe1:8cff/64 Scope:Link
            UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
            RX packets:7642 errors:0 dropped:0 overruns:0 frame:0
            TX packets:1860 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:1000
            RX bytes:1106180 (1.0 MiB)  TX bytes:169633 (165.6 KiB)

[ruib@ruib-laptop ~]$
```



## Configuração multi-plataforma - 4

- alteração do *shell script* de lançamento das máquinas virtuais de Java  
registry\_com.sh

```
[... dir_registry]$ pwd
```

```
/home/ruib/Teaching/java/exemplos/Back_Engine_Alunos/dir_registry
```

```
[... dir_registry]$ ls -l
```

```
total 20
```

```
drwxr-xr-x. 2 ruib ruib 4096 2010-04-24 11:04 interfaces
```

```
-rw-r--r--. 1 ruib ruib  196 2006-04-27 13:38 java.policy
```

```
drwxr-xr-x. 2 ruib ruib 4096 2010-04-24 11:06 registry
```

```
-rwxr-xr-x. 1 ruib ruib  184 2010-04-19 17:04 registry_com_alt.sh
```

```
-rwxr-xr-x. 1 ruib ruib  159 2010-04-19 17:05 registry_com.sh
```

```
[... dir_registry]$ cat registry_com.sh
```

```
java -Djava.rmi.server.codebase="http://192.168.1.68/ruib/classes/"\  
-Djava.rmi.server.useCodebaseOnly=true\  
-Djava.security.policy=java.policy\  
registry.ServerRegisterRemoteObject
```

*softlink para o directório Public*

```
[... dir_registry]$
```

## Configuração multi-plataforma - 5

- alteração do *shell script* de lançamento das máquinas virtuais de Java  
serverSide\_com.sh

```
[... dir_serverSide]$ pwd
```

```
/home/ruib/Teaching/java/exemplos/Back_Engine_Alunos/dir_serverSide
```

```
[... dir_serverSide]$ ls -l
```

```
total 20
```

```
drwxr-xr-x. 2 ruib ruib 4096 2010-04-24 11:40 interfaces
```

```
-rw-r--r--. 1 ruib ruib  196 2006-04-27 13:38 java.policy
```

```
drwxr-xr-x. 2 ruib ruib 4096 2010-04-24 11:48 serverSide
```

```
-rwxr-xr-x. 1 ruib ruib  181 2010-04-19 17:03 serverSide_com_alt.sh
```

```
-rwxr-xr-x. 1 ruib ruib  154 2010-04-19 17:04 serverSide_com.sh
```

```
[... dir_serverSide]$ cat serverSide_com.sh
```

```
java -Djava.rmi.server.codebase="http://192.168.1.68/ruib/classes/"\  
      -Djava.rmi.server.useCodebaseOnly=true\  
      -Djava.security.policy=java.policy\  
      serverSide.ServerComputeEngine
```

*softlink para o directório Public*

```
[... dir_serverSide]$
```

## *Configuração multi-plataforma - 6*

- criação de duas janelas de *shell* para lançamento dos componentes da aplicação residentes na plataforma 1

***Janela 1*** (garantir que o directório base não contém ficheiros \*.class)

```
[... Back_Engine_Alunos]$ cd  
[... Back_Engine_Alunos]$ pwd  
/home/ruib  
[... ~]$ set_rmiregistry 22###
```

***Janela 2***

```
[... dir_registry]$ pwd  
/home/ruib/Teaching/java/exemplos/Back_Engine_Alunos/dir_registry  
[... dir_registry]$ registry_com.sh  
Nome do nó de processamento onde está localizado o serviço de registo? 192.168.1.68  
Número do port de escuta do serviço de registo? 22###  
Stub was generated!  
RMI registry was created!  
RegisterRemoteObject object was registered!
```

## Configuração multi-plataforma - 7

### Janela de shell na plataforma 2

```
[... dir_serverSide]$ pwd
/home/ruib/Teaching/java/exemplos/Back_Engine_Alunos/dir_serverSide
[... dir_serverSide]$ serverSide_com.sh
Nome do nó de processamento onde está localizado o serviço de registo? 192.168.1.68
Número do port de escuta do serviço de registo? 22###
Stub was generated!
RMI registry was created!
ComputeEngine object was registered!
```

### Janela de shell na plataforma 3

```
[... dir_clientSide]$ pwd
/home/ruib/Teaching/java/exemplos/Back_Engine_Alunos/dir_clientSide
[... dir_clientSide]$ clientSide_com.sh 10
Nome do nó de processamento onde está localizado o serviço de registo? 192.168.1.68
Número do port de escuta do serviço de registo? 22###
3.1415926536
[... dir_clientSide]$
```

↗ n.º de algarismos da parte decimal de  $\pi$