

DISTRIBUTED SYSTEMS

Project 1

Nuno Preguiça, Sérgio Duarte, Dina Borrego, João Vilalonga

OBJETIVO

Sistema de ficheiros para armazenar ficheiros usando o espaço livre nas máquinas dos utilizadores.

Propriedades:

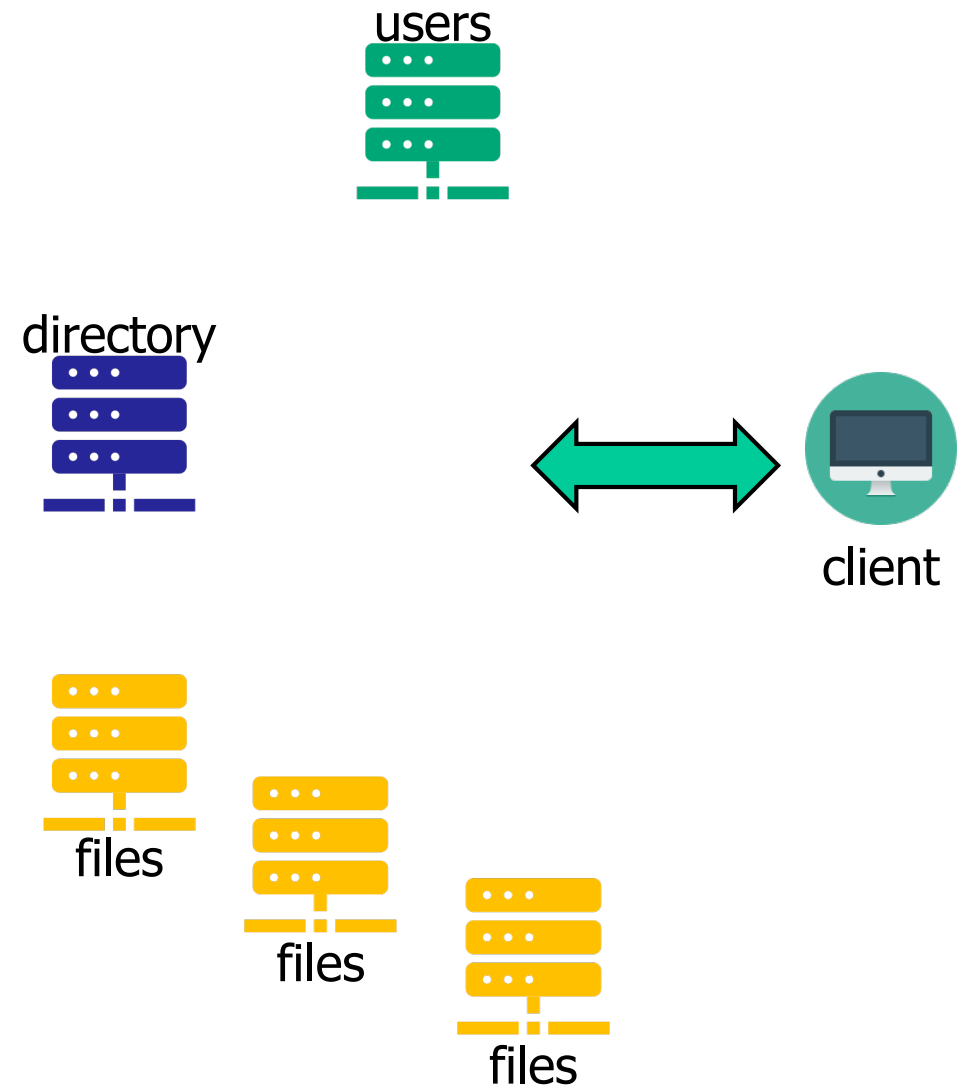
- Sistema usado por múltiplos utilizadores;
- Sistema composto por múltiplas máquinas;
- Utilizadores guardam ficheiros e podem partilhar esses ficheiros com outros utilizadores.

ARQUITETURA

Serviço de utilizadores (users) – mantém informação dos utilizadores do sistema.

Serviço de diretório (directory) – mantém diretório de ficheiros – para cada ficheiro, quem o criou, com quem está partilhado e qual o servidor de ficheiros em que está armazenado.

Serviço de ficheiros (files) – mantém conteúdo dos ficheiros.



EXECUÇÃO: DESCOBERTA DE SERVIÇOS



Os serviços anunciam-se usando multicast. E.g.:
um servidor de utilizadores envia a seguinte mensagem:

```
users http://10.23.0.2:8080/rest/users
```

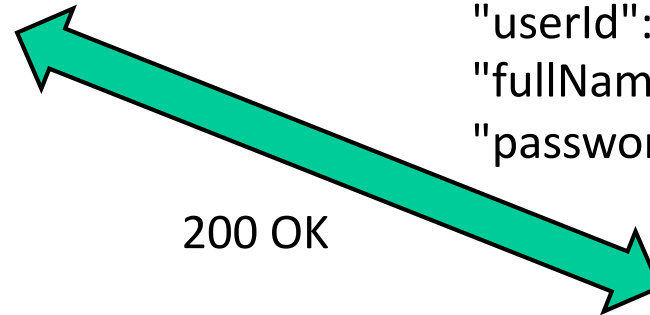
Os clientes e outros servidores usam esta informação para descobrir o URL de um serviço.

EXECUÇÃO: CRIAR UTILIZADOR

POST

`http://users.ourorg:8080/rest/users`

```
{ "email": "jackie.rau@ourorg",  
  "userId": "jackie.rau",  
  "fullName": "Sergio Cartwright",  
  "password": "pcfr3k96" }
```



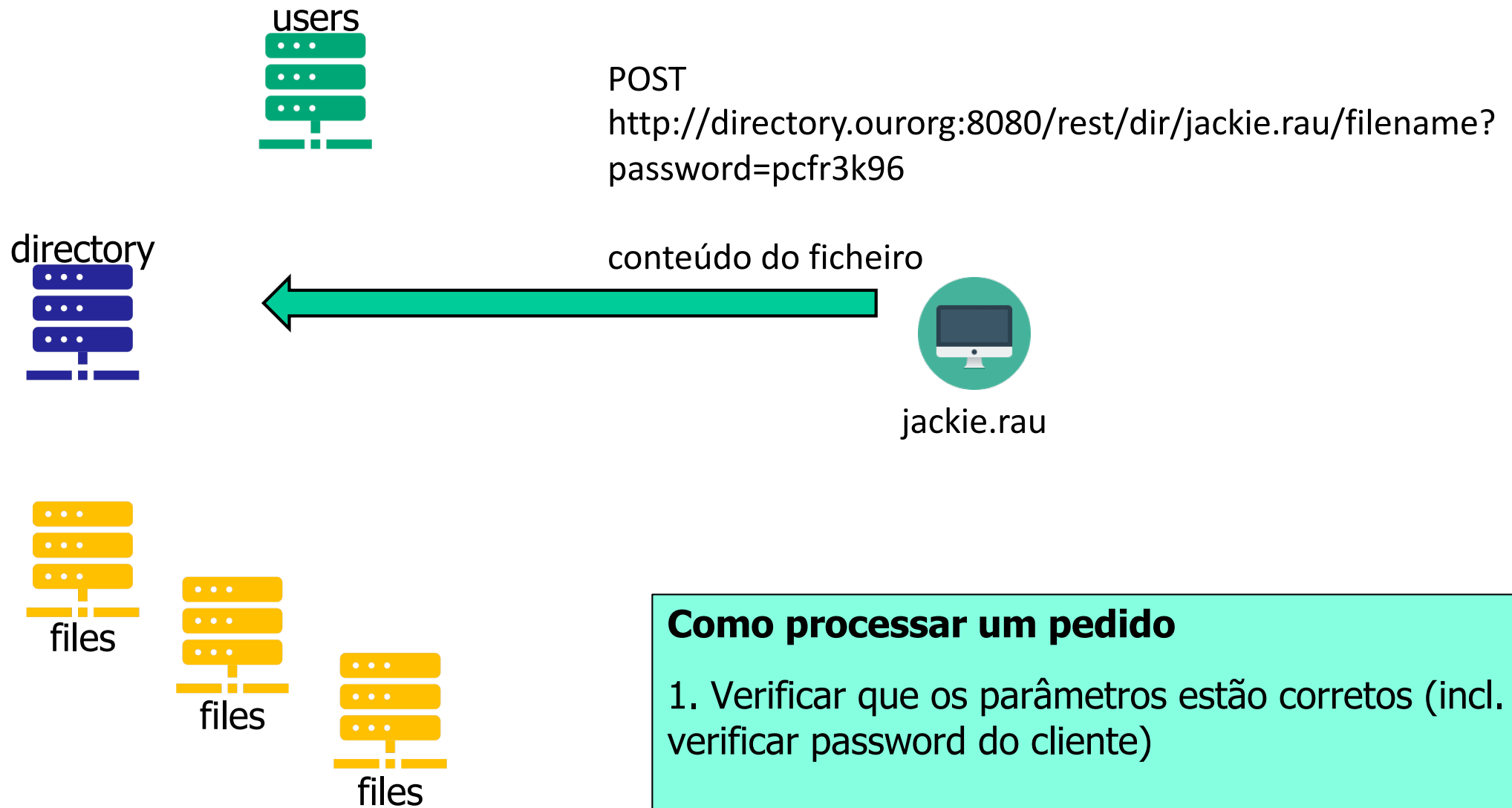
200 OK

"jackie.rau"



jackie.rau

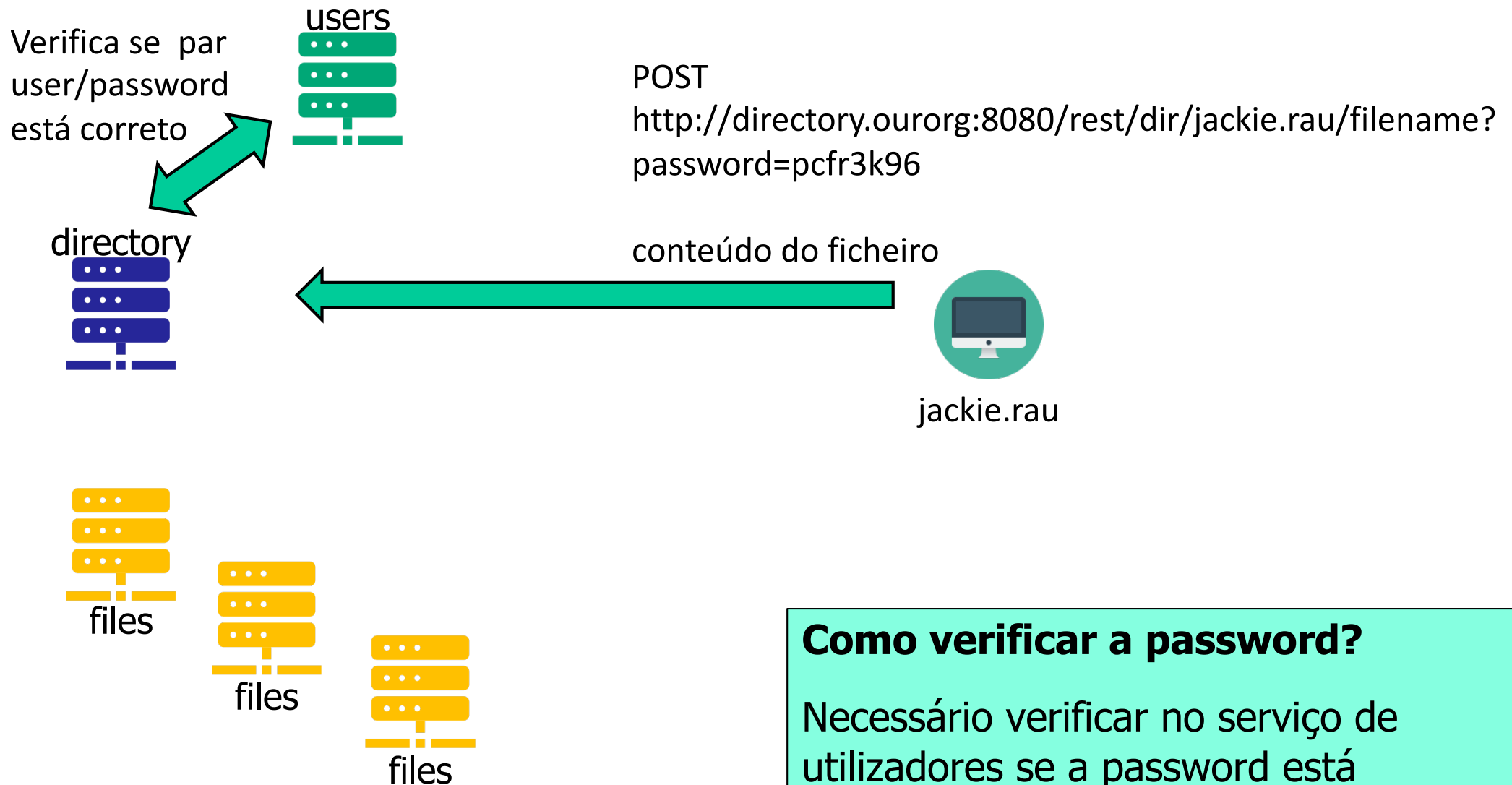
EXECUÇÃO: ESCREVER FICHEIRO (1)



Como processar um pedido

1. Verificar que os parâmetros estão corretos (incl. verificar password do cliente)

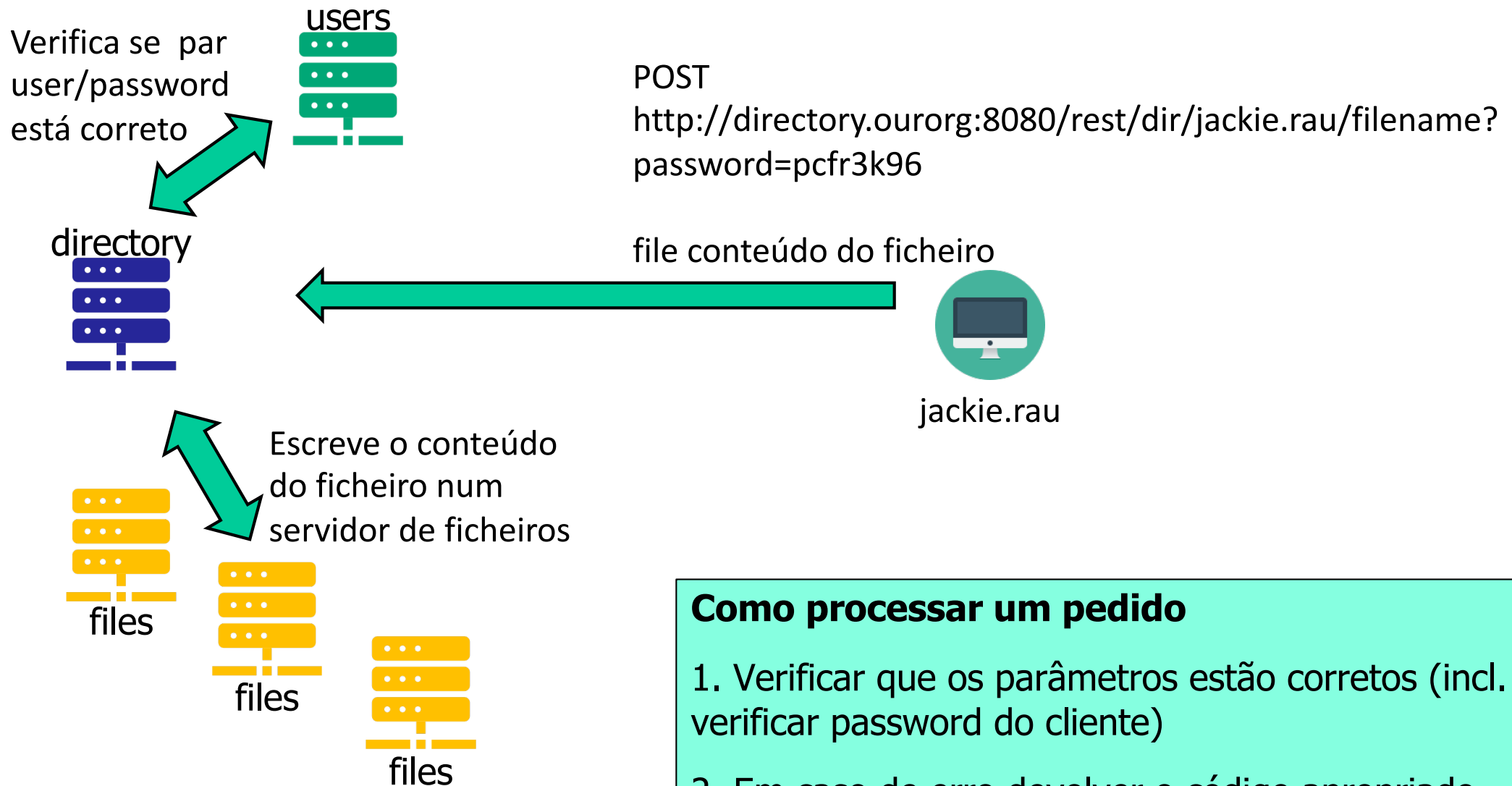
EXECUÇÃO: ESCREVER FICHEIRO (2)



Como verificar a password?

Necessário verificar no serviço de utilizadores se a password está correta. E.g. pode-se usar o *endpoint* getUser.

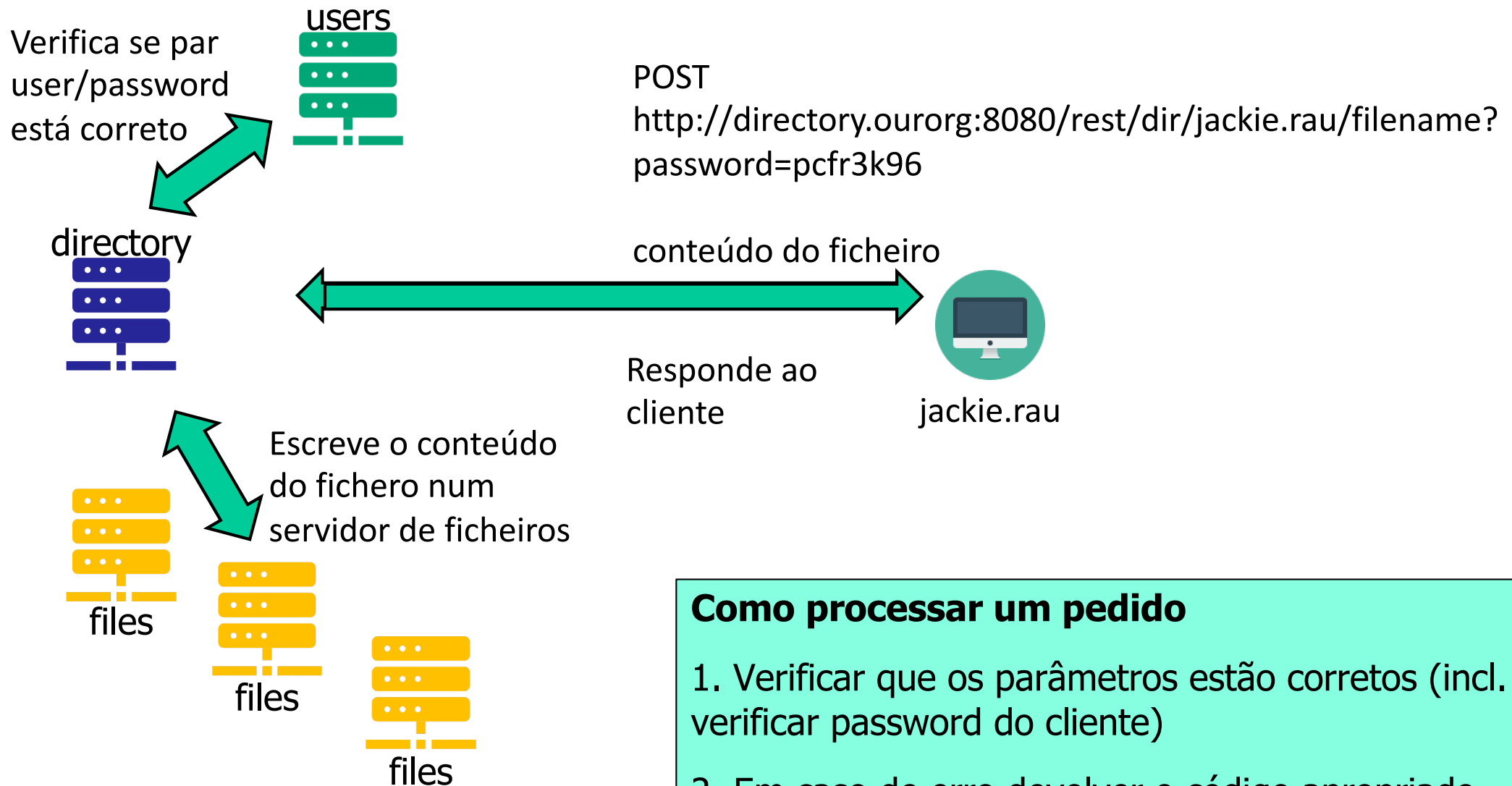
EXECUÇÃO: ESCREVER FICHEIRO (3)



Como processar um pedido

1. Verificar que os parâmetros estão corretos (incl. verificar password do cliente)
2. Em caso de erro devolver o código apropriado. Em caso de sucesso, escreve ficheiro num servidor de ficheiros.

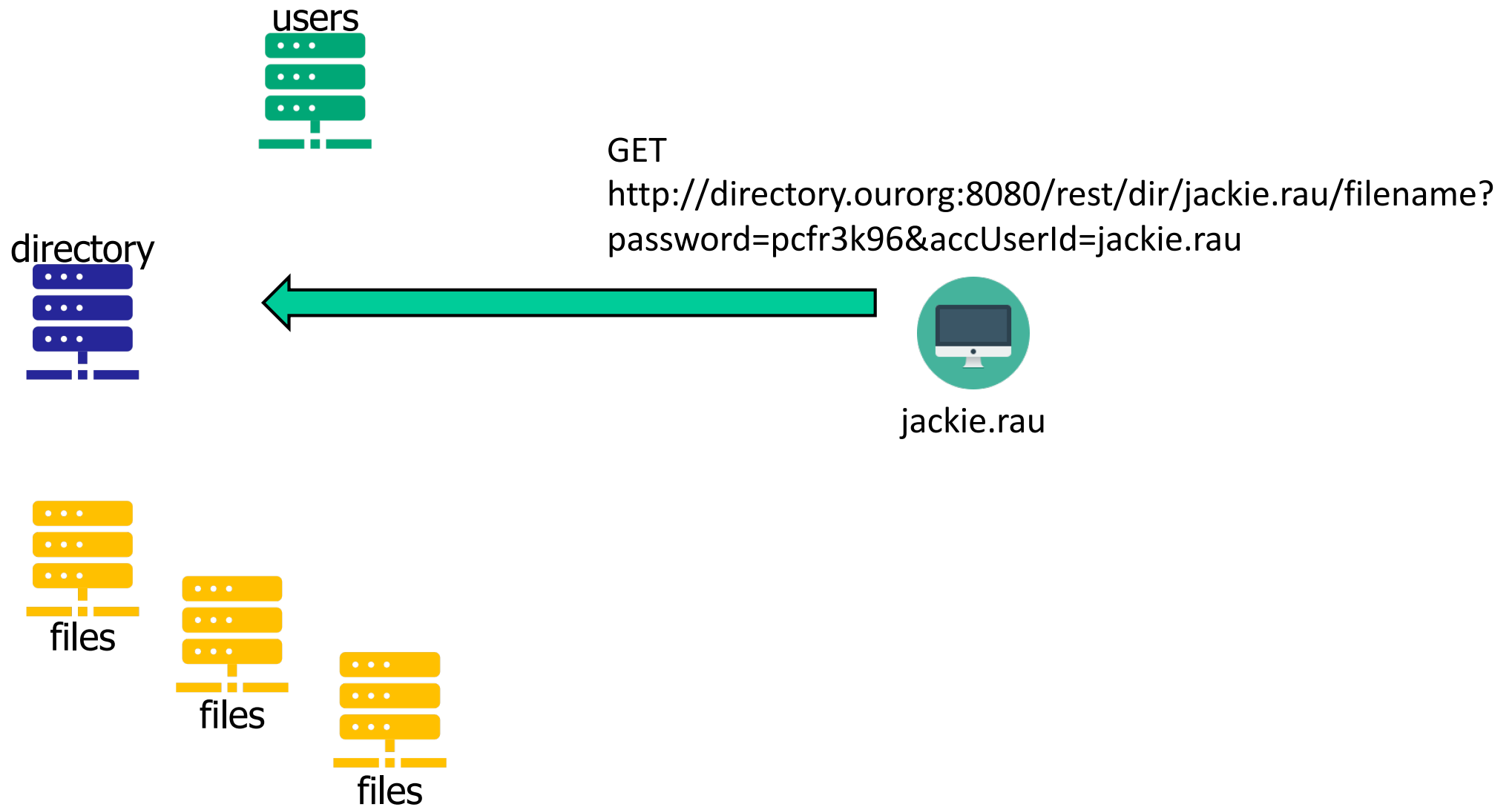
EXECUÇÃO: ESCREVER FICHEIRO (4)



Como processar um pedido

1. Verificar que os parâmetros estão corretos (incl. verificar password do cliente)
2. Em caso de erro devolver o código apropriado. Em caso de sucesso, escreve ficheiro num servidor de ficheiros.

EXECUÇÃO: LER FICHEIRO



EXECUÇÃO: LER FICHEIRO (2)

Verifica se par
user/password
está correto e se
user pode ler
ficheiro



GET

`http://directory.ourorg:8080/rest/dir/jackie.rau/filename?
password=pcfr3k96&accUserId=jackie.rau`



jackie.rau



files



files



files

Quem pode ler um ficheiro

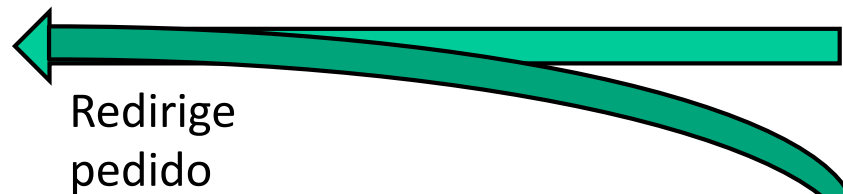
1. Dono do ficheiro (utilizador que criou o ficheiro)
2. Utilizador com quem o ficheiro foi partilhado – o dono do ficheiro pode partilhar o ficheiro com outros utilizadores.

EXECUÇÃO: LER FICHEIRO (3)

Verifica se par
user/password
está correto e se
user pode ler
ficheiro



GET
[http://directory.ourorg:8080/rest/dir/jackie.rau/filename?
password=pcfr3k96&accUserId=jackie.rau](http://directory.ourorg:8080/rest/dir/jackie.rau/filename?password=pcfr3k96&accUserId=jackie.rau)



Redirige
pedido



jackie.rau



Em REST pode-se usar o HTTP redirect para ser mais eficiente. Código:

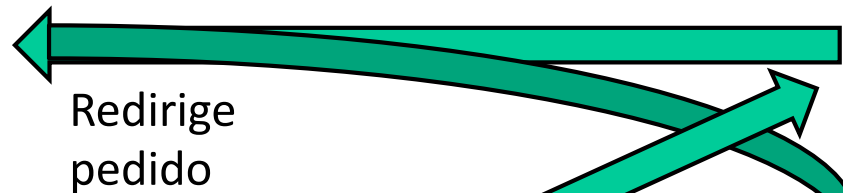
```
throw new WebApplicationException(  
    Response temporaryRedirect(  
        URI.create("https://...")).build());
```

EXECUÇÃO: LER FICHEIRO (4)

Verifica se par
user/password
está correto e se
user pode ler
ficheiro



GET
`http://directory.ourorg:8080/rest/dir/jackie.rau/filename?
password=pcfr3k96&accUserId=jackie.rau`



Redirige
pedido



jackie.rau

Resposta direta
para o cliente



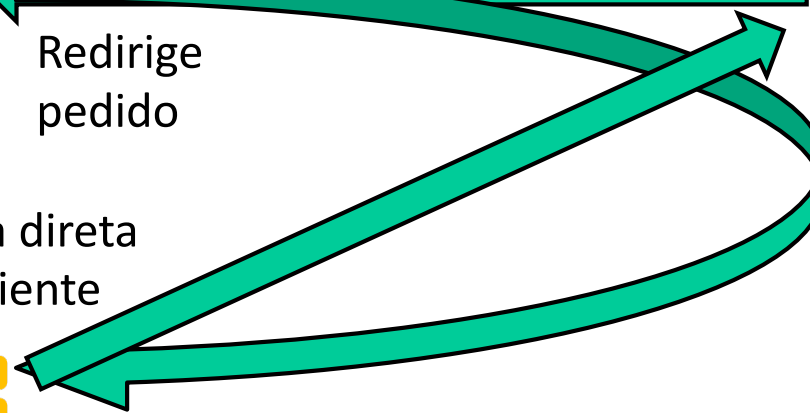
files



files



files



REQUISITOS MÍNIMOS

Requisitos mínimos (max: 8 valores)

API REST - Servidores versão REST funcionais, com todas as operações funcionais quando não há falhas de comunicação;

Auto-configuração - A descoberta por multicast funciona corretamente.

REQUISITOS BASE

Requisitos base (max: 13 valores)

Funcionalidade completa - Servidores REST funcionais, com todas as operações a executar corretamente. Os servidores devem funcionar corretamente mesmo na presença de falhas - com falhas inferiores a 10 segundos, as operações devem executar normalmente; com falhas superiores a 10 segundos, as operações devem usar outro servidor se disponível ou devolver mensagens de erro apropriadas caso tal não seja possível.

Controlo de concorrência - Os servidores REST funcionam corretamente quando vários clientes efetuam pedidos concorrentemente.

ELEMENTOS VALORATIVOS

Eficiência da solução (max: 3 valores) - O sistema deve ser concebido de modo a ser tão eficiente quanto possível, nomeadamente:

- Garantindo uma distribuição de carga entre os servidores de ficheiros, através da distribuição dos ficheiros pelos diferentes servidores de ficheiros;
- Removendo informação (e.g. ficheiros) que não pode ser acedida (e.g. ficheiros que tenham sido criados num servidor de ficheiros e que devido a uma falha tenham acabado por não ser utilizados);
- Executando assíncronamente operações em diferentes servidores.

ELEMENTOS VALORATIVOS

WebServices Soap (max: 3 valores) - O sistema funciona com servidores SOAP apenas.

WebServices Soap interoperáveis (max: 5 valores; alternativa a WebServices Soap) - O sistema funciona com servidores REST e SOAP no mesmo sistema.

MAIS INFORMAÇÃO

As interfaces dos vários serviços estão disponíveis numa ligação presente na página do enunciado.

Podem ser adicionadas novas operações, mas as operações definidas não devem ser modificadas. A implementação deve respeitar a informação presente na documentação, nomeadamente no que diz respeito aos resultados a serem devolvidos em caso de erro.

MAIS INFORMAÇÃO (2)

Existe uma bateria de testes que permite testar o trabalho – ver enunciado do trabalho. Os testes serão adicionados durante o período de execução do trabalho, acompanhando a execução esperada.

- NOTA: passar os testes não garante que o trabalho está correto, dado que não se podem testar todas as situações.

Data de entrega: 24 de abril, 23h59