



UNIVERSIDADE D
COIMBRA

Projeto de Sistemas Distribuídos

ucDrive

Trabalho realizado por:

- João Silva (2019216753)
- Mário Lemos (2019216792)
- Pedro Martins (2019216826)

Índice

Arquitetura do Software	3
Funcionamento do ucDrive	5
Mecanismo de Failover	8
Distribuição de tarefas	11
Testes feitos à plataforma.....	13

Arquitetura do Software

No esquema descrito na **Figura 1** é possível observar a estrutura e arquitetura do modelo cliente-servidor necessário para implementar todas as funcionalidades do ucDrive. Este conta com o uso de múltiplas threads e sockets com o intuito de garantir o bom funcionamento, não bloqueio de ações e transferência correta dos dados a transmitir.

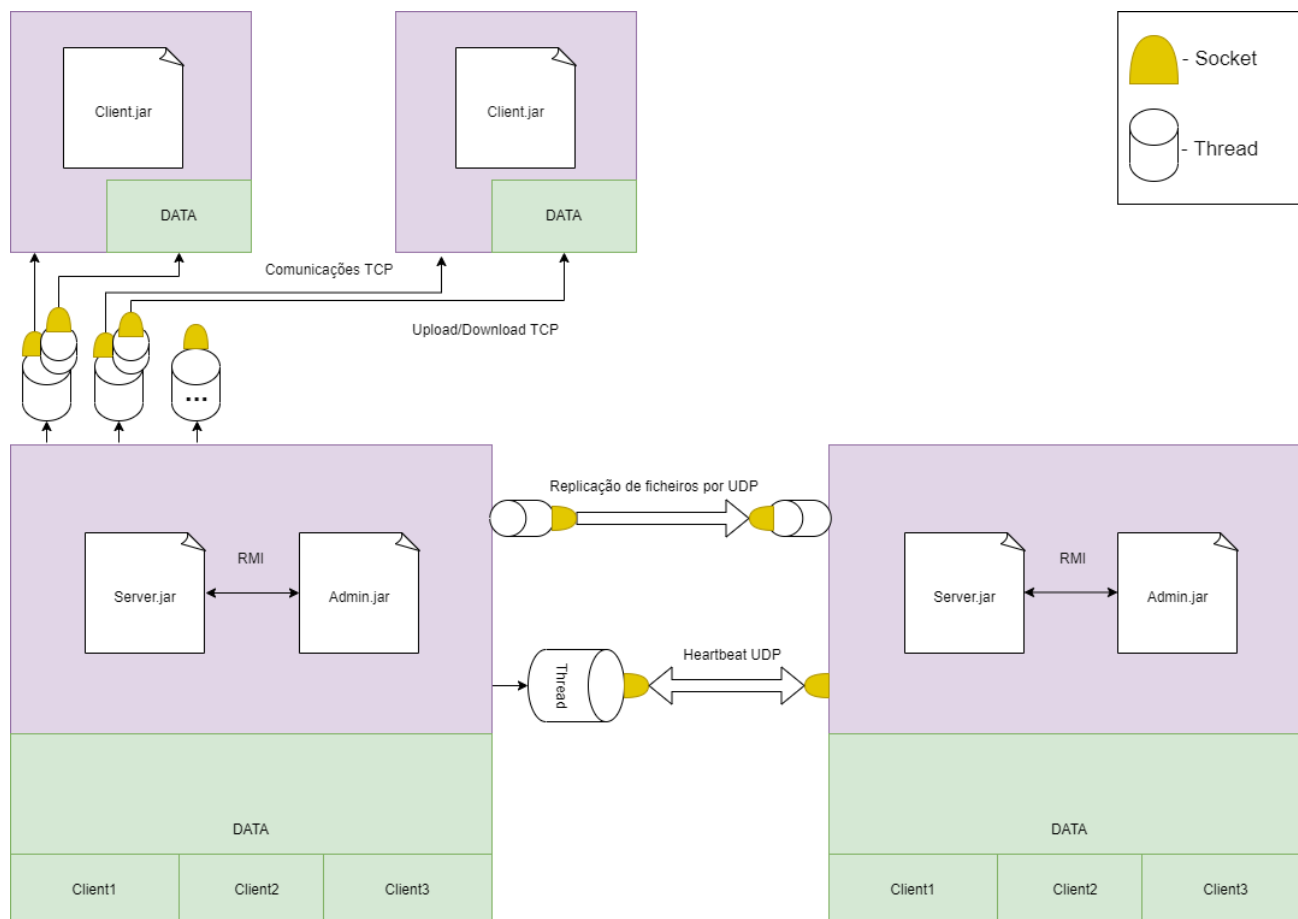


Figura 1 – Detalhes da arquitetura escolhida para implementação do ucDrive

- **Comunicação entre servidor primário e secundário:** Primeiramente é criada uma thread em ambos os servidores primário e secundário de forma a poder garantir uma ação independente na troca de ficheiros por UDP. É também usada a mesma socket para enviar os ficheiros necessários do servidor primário para o secundário, de forma aos dois terem os mesmos ficheiros e, assim, garantir a replicação de dados e que os mesmos se encontram em congruência com as ações de cada utilizador. Além disso o servidor primário cria uma thread com o objetivo de receber e transmitir informação relacionada com os pings heartbeat (explicado em detalhe na parte do Failover).
- **Comunicação entre cliente e servidor:** Ao detetar um novo cliente a conectar-se, o servidor irá em resposta criar uma thread com o objetivo de trabalhar com os comandos fornecidos

pelo utilizador. Esta conexão é feita através de uma socket num porto previamente definido por um administrador. Aquando do download e upload de ficheiros, é criada uma thread propria para tratar deste processo, o que impede o bloqueio do terminal cliente e o deixa prosseguir com outros comandos que o utilizador queira efetuar enquanto espera pelo fim do download/upload. É também aberta uma nova porta num outro porto além do já usado para a comunicação normal para poder assim assegurar a transferência destes dados por pacotes.

- **Comunicação entre servidor e administrador:** Esta comunicação é feita por RMI e é responsável por efetuar diversas configurações no servidor, tais como registro de utilizadores, configurações de Ip e porto e listagem de ficheiros por utilizador e tamanho dos mesmos.

De forma a garantir o bom funcionamento e manutenção de cada um dos servidores, ambos têm de possuir dois ficheiros de texto (users.txt e Configs.txt). O primeiro guarda a informação de cada cliente de forma a esta poder ser usada para efeitos de autenticação no servidor, o segundo contém as configurações de como o servidor deverá operar em certas circunstâncias.

{username}#{pass}#{departamento}#{morada}#{telemóvel}#{identificação}#{diretoria}

Conteúdo por utilizador no ficheiro **users.txt**

{porto primário}

{porto secundário}

{ip primário}

{ip secundário}

{número de pings até assumir estatuto primário}

{frequência dos pings enviados}

Conteúdo das configurações presentes no ficheiro **configs.txt**

Funcionamento do ucDrive

No ucDrive o **cliente** ao iniciar tem duas opções:

- **Iniciar sessão:** o utilizador irá introduzir o seu username e password. Estes dados serão enviados para o servidor de forma a validar os mesmos. Este possui um ArrayList com todos os utilizadores, assim como a password a si associados. Caso exista um utilizador com o nome fornecido e a sua palavra-passe coincida com a recebida, é devolvida uma mensagem de sucesso ao cliente, o que lhe permitirá continuar para o menu com o resto dos comandos. Caso a autenticação falhe, o utilizador terá de reintroduzir os dados de forma a autenticar-se com sucesso.
- **Configurar Ip e porto primário/secundário:** O utilizador pode também configurar os endereços Ip e portos do servidor primário e secundário. Assim, o programa do cliente saberá a que Ip e porto associar a socket criada, de forma a poder estabelecer uma ligação com sucesso ao servidor. É da responsabilidade do cliente assegurar que os ips e portos correspondidos estão corretos na sua aplicação.

De seguida, estão à disponibilidade do **cliente** diversas opções, de entre as quais:

- **Alterar a palavra-passe do utilizador:** O utilizador irá introduzir o comando “changepass” de forma a poder alterar a sua password de autenticação. De seguida, ser-lhe-á pedido a sua palavra passe atual e a sua nova palavra passe duas vezes, de forma a poder confirmar se se encontra bem escrita. Estes dados serão enviados ao servidor que irá confirmar que a palavra atual está bem escrita. Em caso positivo a palavra passe é alterada, sendo transmitido ao cliente uma mensagem de sucesso. Em caso negativo, é devolvido ao utilizador uma mensagem de erro.
- **Listar os ficheiros remotos do cliente no servidor:** Neste método o cliente irá invocar o seguinte comando “lsS”, o qual irá enviar ao servidor um pedido para devolver todos os ficheiros no caminho atual do utilizador no servidor (localização do ficheiro server/Data/{nome_do_cliente}/{...}). Ao receber estes dados o cliente irá apresentar a lista de ficheiros, assim como uma indicação se os mesmos são ficheiros ou diretorias. Em caso de esta estar vazia, simplesmente é imprimida uma mensagem a indicar a falta de ficheiros na diretoria.
- **Listar os ficheiros locais do cliente:** O utilizador irá introduzir o comando “ls”, o que vai buscar diretamente no cliente o seu caminho atual dentro da sua diretoria (localização do ficheiro cliente/Data/{...}). Irá assim ser apresentada a lista de ficheiros devidamente identificada por ficheiros e diretorias. Caso esta se encontre vazia, é devolvida uma mensagem ao cliente.
- **Mudar a diretoria local do cliente:** Neste método o utilizador irá introduzir o comando “cdS {diretoria}”, indicando a diretoria onde quer entrar ou “..” para recuar. Caso o utilizador não tenha permissão para entrar nesta pasta, a diretoria não exista ou o comando seja mal introduzido, uma mensagem de erro é apresentada. Em caso de sucesso o caminho atual do cliente é atualizado com a nova informação.

- **Mudar a diretoria remota do cliente no servidor:** O cliente deverá introduzir o comando “cd {diretoria}”, indicando a diretoria onde quer entrar ou “..” para recuar. Estes dados são enviados para o servidor que em caso de sucesso irá atualizar o caminho do cliente no servidor com a informação fornecida. Caso esta diretoria não exista, o utilizador não tenha acesso à mesma ou o comando tenha sido mal introduzido é devolvido uma mensagem de erro ao cliente.
- **Baixar ficheiros do servidor:** Ao introduzir o comando “downloadS” irá ser pedido ao cliente o nome de um ficheiro a baixar, sendo esta informação fornecida ao servidor. Caso o ficheiro não exista no caminho remoto do cliente, é devolvida uma mensagem de erro ao utilizador. Caso o ficheiro seja encontrado, o mesmo é enviado numa socket independente através de pacotes de 4KB por iteração até o ficheiro ser completamente enviado e recebido pelo cliente. Caso o ficheiro tenha sido recebido na sua totalidade, é devolvida uma mensagem de sucesso, caso contrário é apresentada uma mensagem de erro, que leva à exclusão do ficheiro da pasta local do utilizador.
- **Enviar ficheiros para o servidor:** Neste método o cliente deve introduzir o comando “uploadS” e, de seguida, o nome do ficheiro a enviar para o servidor. Na parte do cliente é verificado se o ficheiro existe no caminho local do utilizador. Caso não exista é devolvida uma mensagem de erro, caso contrário é enviado o ficheiro numa socket independente em pacotes de 4KB por iteração, até o ficheiro ser enviado na sua totalidade. Da parte do servidor, caso o ficheiro recebido esteja completo (não houve perda por parte do cliente) o mesmo fica armazenado no caminho remoto do utilizador no servidor, caso contrário, o ficheiro é descartado e removido, uma vez que não se encontra completo.
- **Configurar Ip e porto primário e secundário:** O utilizador irá introduzir o comando “config” o que lhe permitirá atualizar os Ips e portos do servidor primário e secundário aos quais o cliente se vai conectar. Todas as operações a decorrer serão assim terminadas e o utilizador terá de efetuar novamente a autenticação.
- **Listar os comandos disponíveis no terminal:** Neste método o cliente deve introduzir o comando “help”, o que irá devolver a lista de comandos disponíveis a usar no terminal do utilizador.
- **Sair do terminal:** Ao introduzir o comando “exit” o scanner é encerrado, uma mensagem de despedida é enviada ao servidor de forma a garantir o fecho controlado da socket e thread associada a este cliente e o programa fecha.

Finalmente, também o **administrador** tem acesso aos seguintes métodos, a transmitir por RMI:

- **Registar um utilizador:** Ao introduzir o comando “register” o administrador introduz um username, password e departamento do novo utilizador a inserir. Estes dados são fornecidos ao servidor, que irá por conseguinte criar uma pasta para o cliente na diretoria “/Data” e atualizar o ficheiro de utilizadores (Users.txt).
- **Listar diretorias/ficheiros de cada utilizador:** Neste método o administrador irá introduzir o comando “list”, tendo de seguida fornecer um username válido de um cliente. O servidor irá então devolver a lista de todos os ficheiros e diretorias do utilizador em questão. Caso não existam ficheiros na sua diretoria, devolverá uma mensagem de erro.

- **Configuração das definições de failover:** Aqui o administrador introduz o comando “failover”, o que lhe permitirá configurar o número de pings necessários até considerar um servidor offline e o tempo entre cada ping emitido. Esta informação é passada ao servidor que irá alterar estes valores no seu ficheiro de configurações (Configs.txt).
- **Listar detalhes sobre armazenamento:** Após introduzir o comando “size” o administrador pode ou fornecer um username de um cliente válido ou introduzir a palavra “all”. No primeiro caso, o servidor irá devolver o tamanho total associado à diretoria desse cliente no servidor. No segundo caso, devolve todo o espaço ocupado pelos dados de todos os utilizadores no servidor.
- **Validar a replicação de dados entre os servidores:** Ao introduzir o comando “copy”, é enviado ao servidor um pedido para enviar as diretorias do servidor primário e secundário, de forma a poder comparar as diferenças entre ambas.
- **Listar os comandos disponíveis no terminal:** Neste método o administrador deve introduzir o comando “help”, o que irá devolver a lista de comandos disponíveis a usar no terminal.
- **Sair do terminal:** Ao introduzir o comando “exit” o scanner é encerrado e é emitida uma mensagem de despedida.

Mensagens de erro associadas aos métodos acima mencionados:

- *"Incorrect login credentials, please enter them again!":* Credenciais de autenticação erradas.
- *"There are no files in the current folder!":* O utilizador está a tentar listar ficheiros de uma pasta que se encontra vazia.
- *"The system cannot find the path specified":* O caminho remoto ou local não existe.
- *"You don't have permission to navigate to here":* O utilizador está a tentar entrar numa zona restrita da mudança de caminho quer local quer remoto.
- *"{command} is not recognized as an internal or external command":* O comando inserido não existe no sistema.
- *"The passwords introduced are not equal, please try again.":* As novas palavras passe introduzidas não são idênticas.
- *"Wrong password, please try again.":* A palavra passe atual introduzida não está correta.
- *"File not available in current server directory!":* Ao fazer download, não é possível encontrar o ficheiro no caminho atual do servidor.
- *"The required file doesn't exist in the current directory!":* Ao fazer upload, não é possível encontrar o ficheiro no caminho atual do cliente.
- *"Failure while downloading a file! Corrupted file has been deleted":* O ficheiro baixado está corrompido e foi apagado com sucesso.
- *"Failure while downloading a file! Corrupted file cannot be deleted":* O ficheiro baixado está corrompido, mas não foi possível apagá-lo.
- *"Failure while uploading a file! File might be corrupted!":* Ocorreu um problema a enviar um ficheiro para o servidor. Alerta para possível corrupção do ficheiro na diretoria de destino.
- *"There has been a connection error. Please input the desired command again!":* O servidor primário está em baixo e, como tal, não é possível de momento estabelecer conexão.

Mecanismo de Failover

Neste sistema o servidor secundário estabelece um sistema de heartbeat, no qual através de UDP irá enviar pings a cada unidade de tempo (definida pelo administrador) e se o primário não responder após um número de pings (também definido pelo administrador) o secundário irá assumir o papel de primário. Em startup e após o servidor primário receber um ficheiro, este terá de atualizar as diretorias do servidor secundário, garantindo que o mesmo se encontra atualizado.

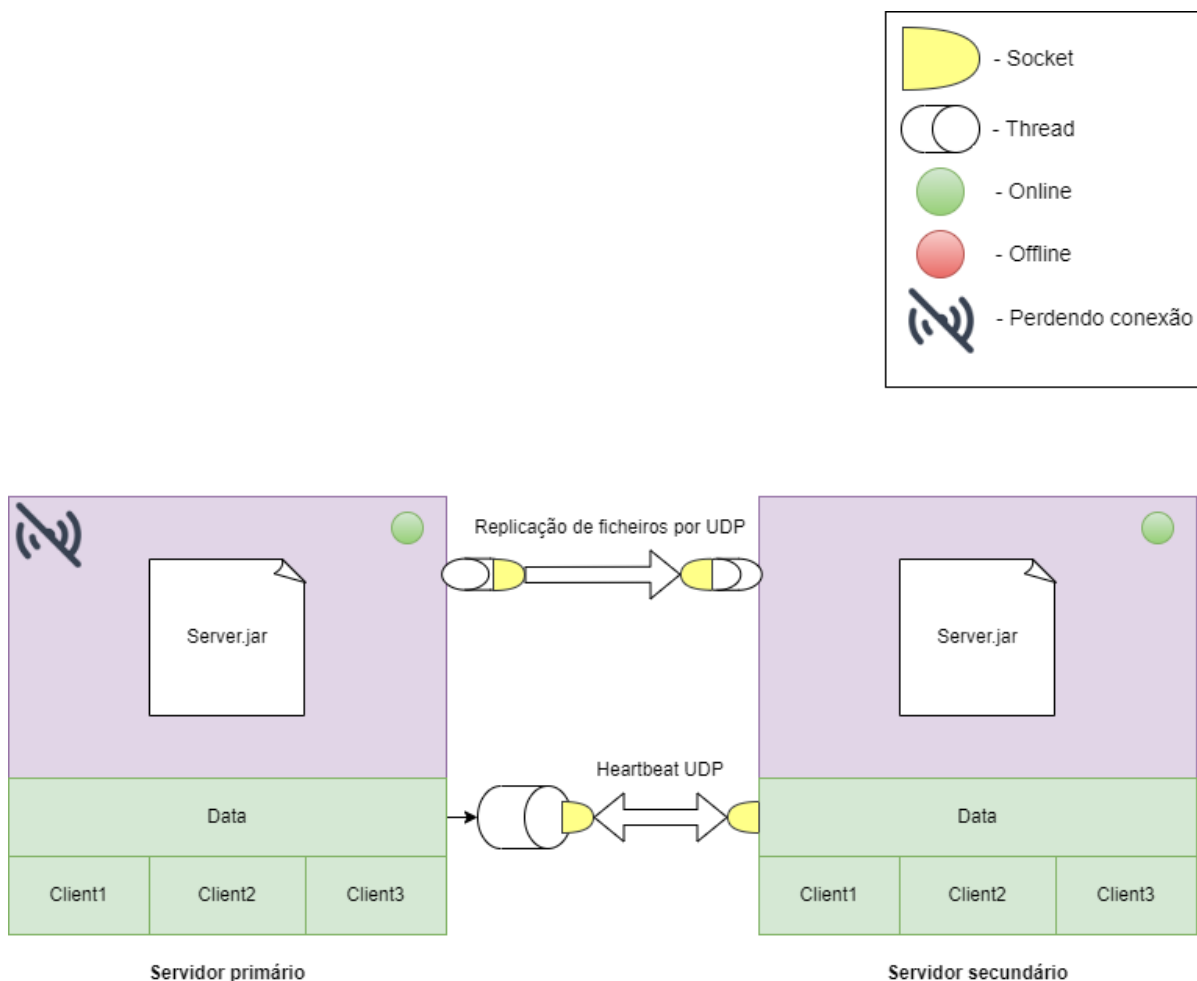


Figura 2 – Início da perda de comunicação por parte do servidor primário

Na **Figura 2** demonstra-se a comunicação entre o servidor primário e secundário, num cenário em que o primário começa a perder a conexão, o que em troca faz com que o secundário não receba resposta dos pings que efetua. Quando este número de pings falhado excede o limite previamente estabelecido, o servidor secundário irá passar a assumir o papel de primário. Enquanto este não for primário sempre que o cliente tentar efetuar uma comunicação com o servidor, que o alertará para problemas de conexão, devolvendo-o ao menu correspondente. Quando este finalmente se consegue associar ao secundário (agora primário), antes de mandar qualquer método (tal como ls) irá enviar de novo as credenciais previamente guardadas (username e palavra-passe) de forma a

poder-se autenticar. Esta solução não só impede o utilizador de voltar a introduzir as suas credenciais, como também garante o menor tempo offline necessário e eficiência na troca de servidores.

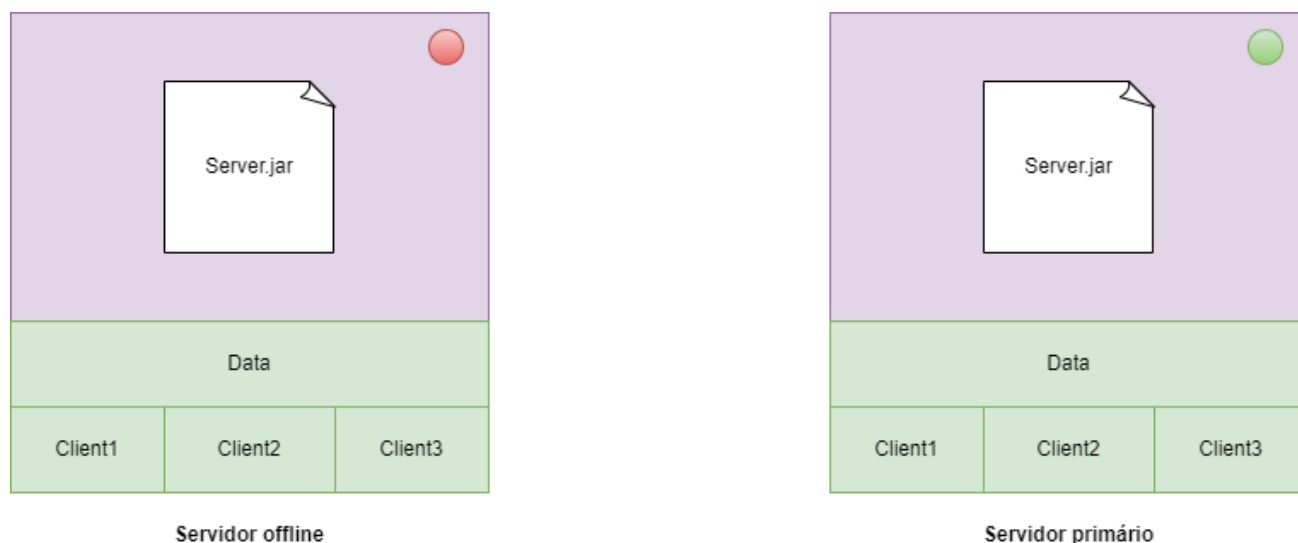


Figura 3 – Estado atual, com o antigo servidor primário offline

A **Figura 3** demonstra que o ex-servidor secundário (agora primário) se encontra online e a responder a todos os clientes e administradores que pretendam comunicar com ele.

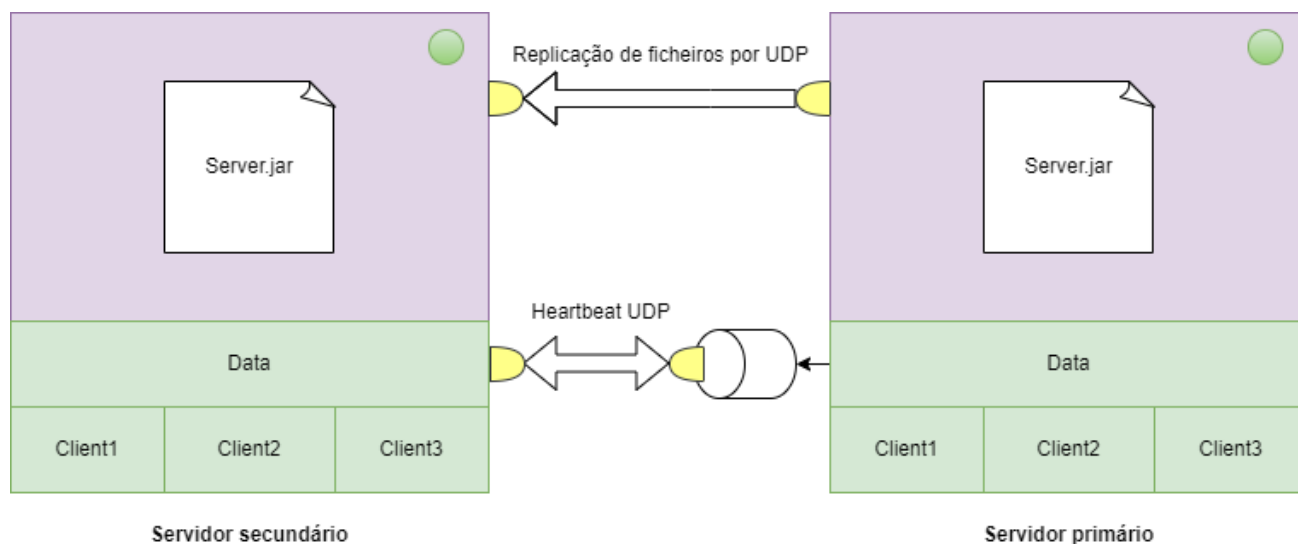


Figura 4 – Antigo servidor primário volta a ficar online, agora é secundário

Tal como demonstra a **Figura 4**, o antigo servidor primário voltou a ficar online e assumiu o papel de secundário, ficando assim responsável por enviar os pings até o servidor primário falhar e assumir assim esse lugar, usando os mecanismos previamente descritos.

Replicação dos dados:

Esta parte é crucial para garantir uma boa sincronização e congruência dos dados entre os servidores primário e secundário. Como é ilustrado na **Figura 5**, cada servidor é responsável por criar uma thread. Do lado do secundário, essa thread é criada aquando do processo de heartbeat, após 10 pings bem sucedidos. O seu objetivo é esperar por ficheiros que sejam recebidos do lado do primário, ou simplesmente recebê-los na totalidade. Do lado do servidor primário, esta thread responsável por enviar ficheiros ao secundário é criada em diversos momentos do código, tais como:

- **Upload:** Ao receber um ficheiro do cliente, o servidor primário é responsável por enviar o mesmo ao secundário, atualizando assim a sua informação de ficheiros.
- **Register:** Quando um administrador adiciona um utilizador ao ucDrive, esta informação é atualizada na parte do registro no servidor (ficheiro users.txt previamente explicado). Aqui será criada uma thread, de forma a garantir que o secundário tem a lista de cliente mais atual.
- **Config:** Quando um administrador altera alguma configuração do servidor, esta é atualizada no método de failover do servidor (ficheiro configs.txt previamente explicado). Isto levará à criação de uma thread responsável por enviar estas atualizações ao secundário, garantindo as configurações mais recentes.
- **List:** Aqui, irá ser feita a verificação de todos os ficheiros, garantindo a ambiguidade de ambas as diretorias dos dois servidores.

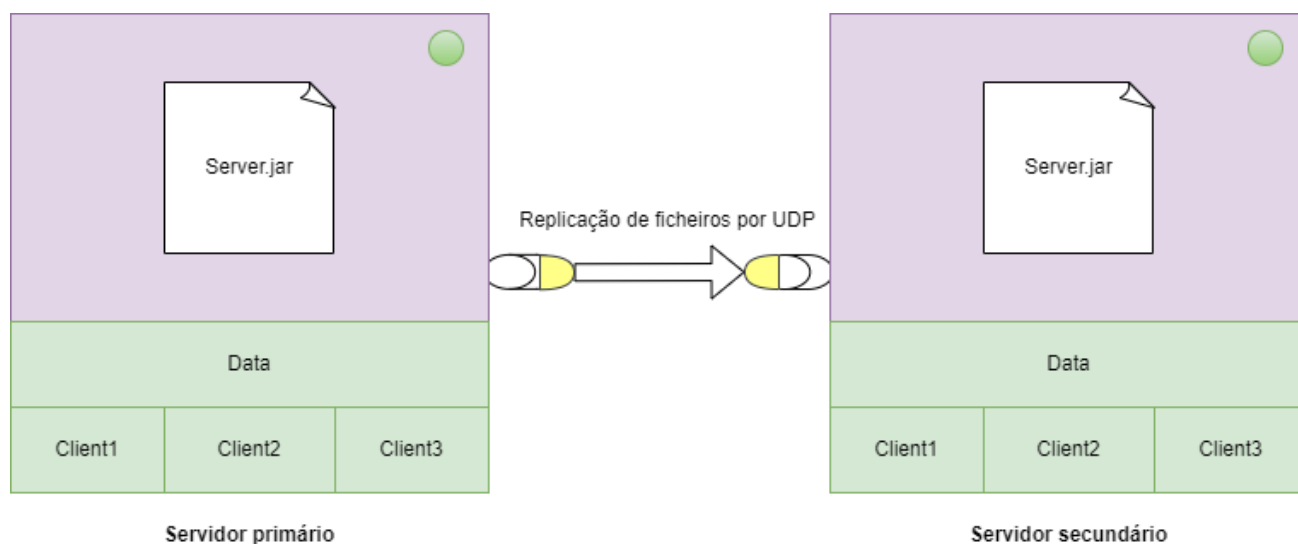


Figura 5 – Mecanismo de replicação dos ficheiros, via UDP

Distribuição de tarefas

Numa fase inicial de desenvolvimento, o grupo inteiro reuniu-se múltiplas vezes para discutir qual a direção a seguir e quais as tecnologias a usar. Como primeira etapa de desenvolvimento foram estas as distribuições efetuadas (é de notar que a estrutura a seguir usada para denotar as tarefas é equivalente àquela fornecida pelo professor com a distribuição da nota incluída, Ids das tarefas são coincidentes):

ID	Descrição da tarefa	João	Mário	Pedro
2.1	Estrutura de comunicação TCP e troca de mensagens.			✓
2.2	Autenticar o do utilizador no terminal de cliente.	✓		✓
2.3	Alterar a password do utilizador.	✓		
2.4	Configurar endereços e portos de servidores primário e secundário.		✓	
2.5	Listar os ficheiros que existem na diretoria atual do servidor.			✓
2.6	Mudar a diretoria atual do servidor.	✓		
2.7	Listar os ficheiros que existem na diretoria atual do cliente.			✓
2.8	Mudar a diretoria atual do cliente.	✓		
2.9	Descarregar um ficheiro do servidor.			✓
2.10	Carregar um ficheiro para o servidor.			✓

Após esta primeira etapa se encontrar concluída, todos os membros do grupo realizaram múltiplos testes e discutiram diferentes abordagens aos problemas encontrados de forma a otimizar o código e melhorar o desempenho e funcionalidade do mesmo. De seguida, seguem-se as etapas do RMI, as quais ficaram quase inteiramente ao cargo de um aluno.

ID	Descrição da tarefa	Mário
1.1	Estrutura de comunicação RMI (Interface, registo no registry: bind e lookup).	✓
1.2	Registar utilizador.	✓
1.3	Listar as directorias/ficheiros por utilizador.	✓
1.4	Configurar o mecanismo de failover.	✓
1.5	Listar detalhes sobre o armazenamento.	✓
1.6	Validar a replicação dos dados entre os vários servidores.	✓

Em simultâneo com o trabalho RMI acima descrito, os outros dois elementos do grupo focaram-se no tratamento de exceções, integração do código e desenvolvimento do segundo servidor, tal como descrito na seguinte tabela:

ID	Descrição da tarefa	João	Mário	Pedro
3.1	Tratamento de exceções.	✓	✓	✓
3.2	Failover - Estrutura de comunicação UDP.	✓	✓	✓
3.3	Failover - Heartbeat e deteção de falha no servidor primário.			✓
3.4	Failover - Ficheiros carregados no primário são copiados para o secundário.	✓	✓	
3.5	Failover – Cliente conecta-se com secundário em caso de avaria no primário	✓		✓
NA	Integração e junção do código desenvolvido.	✓		

Estas duas últimas tabelas representam a segunda etapa do nosso projeto (pós comandos iniciais cliente/servidor) e foram desenvolvidas em simultâneo, de forma a gerir melhor o tempo dispendido. O último ponto com o id **NA** corresponde à integração dos vários métodos desenvolvidos de forma a obter uma única arquitetura funcional. Por fim temos as partes finais do projeto, mais concretamente a elaboração do relatório e testes adicionais de funcionamento integral de todos os sistemas implementados.

ID	Descrição da tarefa	João	Mário	Pedro
4.1	Descrição da arquitetura de software.			✓
4.2	Detalhes sobre o funcionamento do servidor ucDrive.			✓
4.3	Descrição do mecanismo de failover.			✓
4.4	Descrição dos testes feitos à plataforma.			✓
5.1	Atenção aos detalhes (qualidade do código).	✓	✓	✓
NA	Testes aos sistemas desenvolvidos.	✓	✓	✓

Por fim, esta tabela acenta no relatório da atividade, incluindo descrição da arquitetura, sistemas implementados, divisão de tarefas e testes efetuados. O ponto com id **NA** corresponde aos testes efetuados no sistema, os quais tiveram o envolvimento de todos os membros tanto no fim da concepção dos métodos, mas também no produto final obtido.

Testes feitos à plataforma

No final do desenvolvimento da arquitetura e o respetivo código, o foco final foi desenvolver e aplicar diversos testes, com o intuito de perceber as capacidades e limitações do produto criado, resolver possíveis problemas que apareçam e melhorar certas áreas da plataforma conforme necessário.

ID	Descrição do teste	✓/✗
1	Utilizador autentica-se com credenciais certas	✓
2	Utilizador não consegue entrar com credenciais erradas	✓
3	Utilizador altera Ips e portos antes de estar autenticado	✓
4	Utilizador tenta listar os ficheiros da diretoria local	✓
5	Utilizador tenta listar os ficheiros da diretoria remota	✓
6	Utilizador tenta mudar a diretoria local de forma válida	✓
7	Utilizador tenta mudar a diretoria local para lugar restrito e é devolvido ao menu	✓
8	Utilizador tenta mudar a diretoria local para uma inexistente e é devolvido ao menu	✓
9	Utilizador tenta mudar a diretoria remota de forma válida	✓
10	Utilizador tenta mudar a diretoria remota para lugar restrito e é devolvido ao menu	✓
11	Utilizador tenta mudar a diretoria remota para uma inexistente e é devolvido ao menu	✓
12	Utilizador tenta listar um ficheiro local ou remoto não existente e é devolvido ao menu	✓
13	Na configuração de Ips, o utilizador introduz um ip inválido e a operação não prossegue	✓
14	Na configuração de portos, o utilizador introduz um porto inválido e a operação não prossegue	✓
15	O utilizador consegue configurar ips e portos dos servidores a conectar-se com sucesso	✓
16	Ao digitar “help” no terminal, o utilizador é apresentado com a lista de comandos	✓
17	O utilizador consegue enviar um ficheiro da diretoria local para o servidor	✓
18	O utilizador ao fazer upload, se introduzir um ficheiro não existente na diretoria local, é devolvido ao menu com mensagem de erro	✓
19	Caso o terminal do cliente termine a meio do upload para o servidor, este apaga com sucesso o ficheiro, uma vez que o mesmo não foi recebido na sua totalidade	✓
20	O utilizador consegue baixar um ficheiro da sua diretoria remota no servidor	✓
21	O utilizador ao fazer download, se introduzir um ficheiro não existente na diretoria remota, é devolvido ao menu com mensagem de erro	✓
22	Caso o terminal do servidor termine a meio do download do ficheiro, o cliente apaga com sucesso o ficheiro, uma vez que o mesmo não foi recebido na sua totalidade	✓
23	Possibilidade de ter dois clientes a comunicar com o servidor ao mesmo tempo	✓
24	Possibilidade de ter três clientes a comunicar com o servidor ao mesmo tempo	✓
25	Caso o utilizador digite “exit”, a sua conexão com o servidor é corretamente terminada e o terminal fecha com sucesso	✓
26	Caso o utilizador termina abruptamente o terminal (ou este deixe de funcionar), o servidor consegue fechar com sucesso a comunicação com este	✓

27	Utilizador introduz um comando inválido e é devolvido ao menu, com mensagem de erro	✓
28	Utilizador tenta alterar a password, introduzindo a sua atual correta e uma nova válida	✓
29	Utilizador tenta alterar a password, introduzindo a sua atual incorreta, é devolvido ao menu com mensagem de erro	✓
30	Utilizador tenta alterar a password, introduz a sua atual correta contudo introduz uma nova palavra passe inválida, sendo devolvido ao menu com mensagem de erro	✓
31	Utilizador tenta alterar a password, contudo ao introduzir a nova palavra-passe duas vezes, elas não coincidem, levando o cliente ao menu com mensagem de erro	✓
32	Utilizador ao listar os ficheiros locais de uma diretoria vazia, é apresentada uma mensagem de erro, devolvendo-o ao menu	✓
33	Utilizador ao listar os ficheiros remotos de uma diretoria vazia no servidor, é apresentada uma mensagem de erro, devolvendo-o ao menu	✓
34	O utilizador manda uma instrução ao servidor, contudo este encontra-se offline, o que leva o cliente a apresentar uma mensagem de perda de conexão e a tentar conectar-se ao novo servidor secundário (próximo primário)	✓
35	Quando um servidor é iniciado, este verifica se existe algum servidor já online	✓
36	Em início do servidor, caso já exista algum servidor online, este fica em espera (como secundário), enviando pings periódicos para verificar o estado do primário	✓
37	Possibilidade de ter dois servidores como secundários ligados, em caso de falha do primário, um torna-se primário e o outro termina	✓
38	Quando o servidor primário ficar offline, o secundário após um número de pings falhados torna-se no primário.	✓
39	Caso o servidor primário feche abruptamente, todas as conexões são devidamente fechadas	✓
40	O administrador consegue registar um novo utilizador no ucDrive	✓
41	O novo utilizador registado é adicionado com sucesso aos utilizadores do servidor	✓
42	O administrador apresenta os ficheiros de um utilizador válido	✓
43	Ao adicionar um utilizador, o ficheiro de utilizadores atualizado é enviado por UDP para o servidor secundário	✓
44	O administrador consegue alterar as configurações de failover no ficheiro configs.txt	✓
45	As alterações das configurações são corretamente lidas e aplicadas pelo servidor primário	✓
46	Ao alterar as configurações, o ficheiro de configurações atualizado é enviado por UDP para o servidor secundário	✓
47	O administrador consegue ver o tamanho ocupado por cada utilizador no servidor	✓
48	O administrador consegue ver o tamanho total de todos os ficheiros presentes no servidor	✓
49	O administrador consegue ver uma lista de comandos disponíveis no seu terminal, através do comando "help"	✓
50	Ao introduzir o comando "list" o administrador consegue verificar os ficheiros de um utilizador válido	✓
51	Na listagem dos utilizadores pelo administrador, é verificado se os ficheiros encontram replicados nos dois servidores	✓
52	Caso o administrador introduza um comando inválido, deve ser devolvido ao menu, com mensagem de erro	✓
53	No upload de ficheiros, múltiplos ficheiros são passados e encontram-se completos na integra	✓

54	No download de ficheiros, múltiplos ficheiros são baixados e encontram-se completos na íntegra	✓
55	Na replicação de ficheiros entre servidores, múltiplos ficheiros são passados e encontram-se completos na íntegra	✓
56	Os servidores conseguem ler o ficheiro de utilizadores e guardar essa informação	✓
57	Os servidores conseguem ler o ficheiro de configurações e atualizar os mecanismos de failover consoante os valores aqui lidos	✓
58	O servidor consegue ter múltiplos administradores online	✓
59	O servidor primário consegue responder aos pings do secundário enquanto online	✓
60	Ao introduzir o comando “list” seguido da keyword “all” o administrador consegue verificar os ficheiros do servidor	✓
61	O administrador ao efetuar o comando “copy” verifica corretamente os ficheiros em ambos os servidores primário e secundário	✓
62	O administrador ao introduzir um novo utilizador, caso este já se encontre no ficheiro de utilizadores, não é adicionado, apresentando mensagem de erro	✓