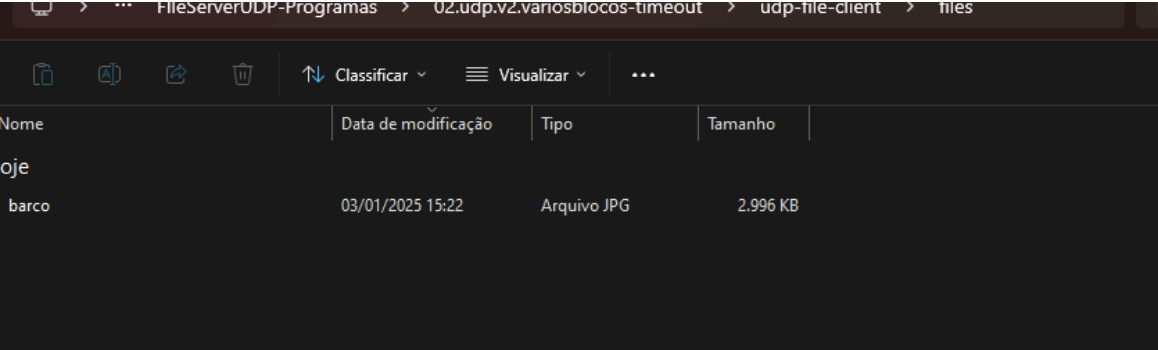


Porque os programas não enviam arquivos grandes como deveriam?

## I- Programa 1: Utilizando timeout

Testando o programa utilizando timeout no cliente que estava configurado com dois segundos, notamos que ao enviar um arquivo superior a 2 MB, o programa cliente não conseguia receber o arquivo completamente, como pode ser visto na Figura 1 e 2. O motivo disso ocorrer é justamente por conta do tempo do timeout ser relativamente baixo fazendo com que se for enviado arquivos grandes, eles não serão enviados corretamente como na Figura 3, onde está localizado o arquivo original.

Figura 1 - Arquivo recebido pelo cliente possuindo 2.996 KB.

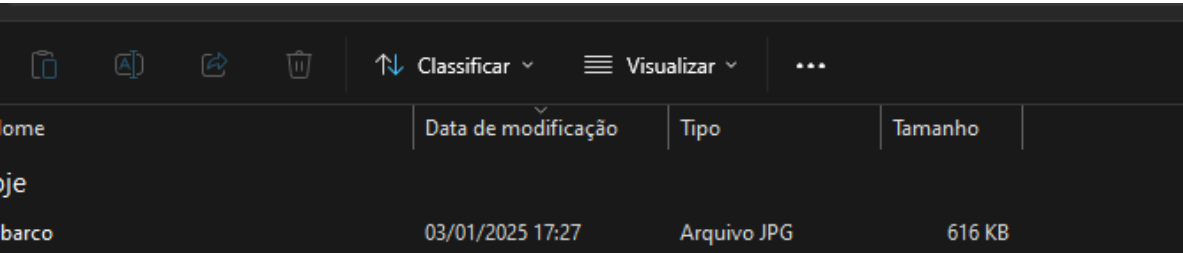


The screenshot shows a file explorer window with the address bar displaying the path: FileServerUDP-Programas > 02.udp.v2.variosblocos-timeout > udp-file-client > files. The toolbar includes icons for file operations and a menu with 'Classificar' and 'Visualizar'. The table below lists the files in the directory.

Nome	Data de modificação	Tipo	Tamanho
barco	03/01/2025 15:22	Arquivo JPG	2.996 KB

Fonte: Autoria Própria, 2025.

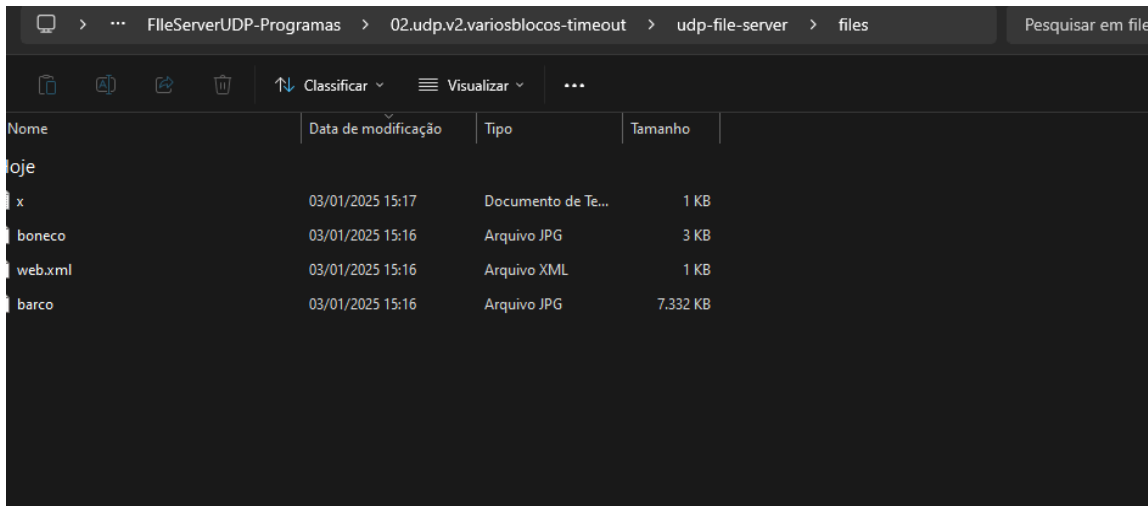
Figura 2 - Arquivo recebido pelo cliente possuindo 616 KB.



The screenshot shows a file explorer window with the address bar displaying the path: FileServerUDP-Programas > 02.udp.v2.variosblocos-timeout > udp-file-client > files. The toolbar includes icons for file operations and a menu with 'Classificar' and 'Visualizar'. The table below lists the files in the directory.

Nome	Data de modificação	Tipo	Tamanho
barco	03/01/2025 17:27	Arquivo JPG	616 KB

Figura 3 - Arquivo barco.jpg no servidor possuindo 7.332 KB.



Nome	Data de modificação	Tipo	Tamanho
oje			
x	03/01/2025 15:17	Documento de Te...	1 KB
boneco	03/01/2025 15:16	Arquivo JPG	3 KB
web.xml	03/01/2025 15:16	Arquivo XML	1 KB
barco	03/01/2025 15:16	Arquivo JPG	7.332 KB

Fonte: Autoria Própria, 2025.

O servidor de arquivos recebe uma solicitação do cliente para enviar o arquivo ‘barco.jpg’ que o envia como representado na Figura 4.

Figura 4: Resposta do servidor ao pedido do cliente.

```
C:\Users\JORDAN\Downloads\FileServerUDP-Programas\02.udp.v2.variosblocos-timeout\udp-file-server> python udp-file-server.py
Escutando em ... ('127.0.0.1', 12345)
Recebi pedido para o arquivo barco.jpg
Enviando arquivo barco.jpg
```

Fonte: Autoria Própria, 2025.

O cliente recebe o arquivo ‘barco.jpg’ que é gravado na sua pasta de arquivos. Entretanto, o arquivo saiu com timeout, ou seja, o tamanho do arquivo ‘barco.jpg’ não foi enviado completamente, pois o timeout estabelecido interrompeu o restante do envio de dados por ter passado do tempo previamente estabelecido no programa cliente como mostrado na Figura 5.

Figura 5: Recebimento de arquivo incompleto no cliente.

```
C:\Users\JORDAN\Downloads\FileServerUDP-Programas\02.udp.v2.variosblocos-timeout\udp-file-client>python udp-file-client.py
Arquivo a pedir ao servidor: barco.jpg
Enviando pedido a ('127.0.0.1', 12345) para barco.jpg

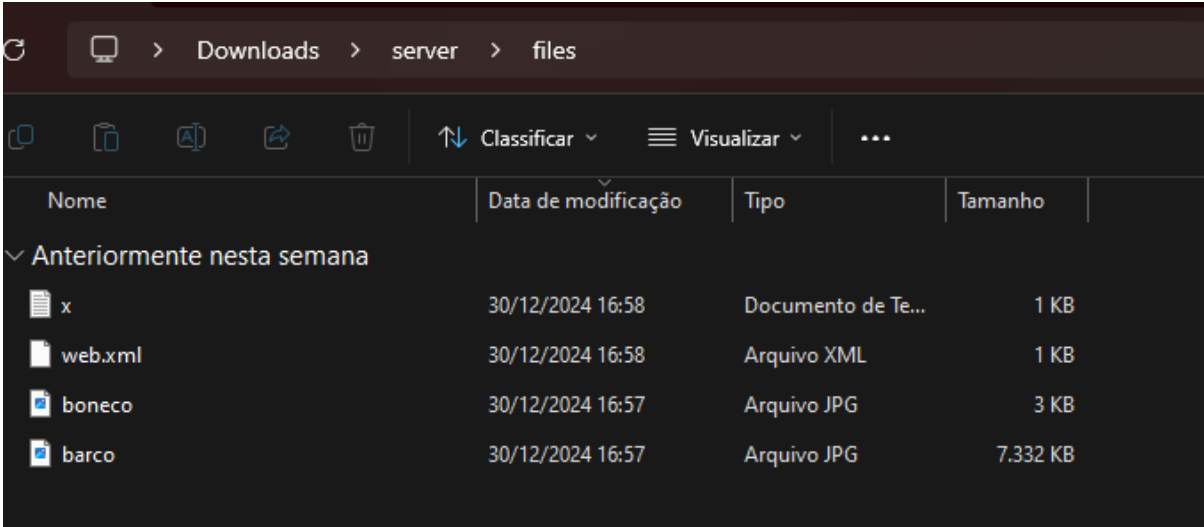
Gravando arquivo localmente
Saiu com timeout ...
Arquivo a pedir ao servidor: |
```

Fonte: Autoria Própria, 2025.

## II - Programa 2: Utilizando tamanho (Em blocos de 1024 e 2048)

Como solicitado, criamos um programa parecido com o primeiro, mas com a diferença de utilizar tamanho com blocos de bytes ao invés do timeout. Testamos solicitar e enviar arquivos com dois blocos de tamanhos diferentes que foram respectivamente 1024 e 2048 Bytes. Em ambos os casos, para arquivos de tamanhos inferiores a 2 MB, foram enviados e recebidos tranquilamente como se pode notar nas Figuras 6 e 7. O problema ocorreu quando tentamos enviar um arquivo muito maior representado pelo 'barco.jpg' que explicaremos a seguir.

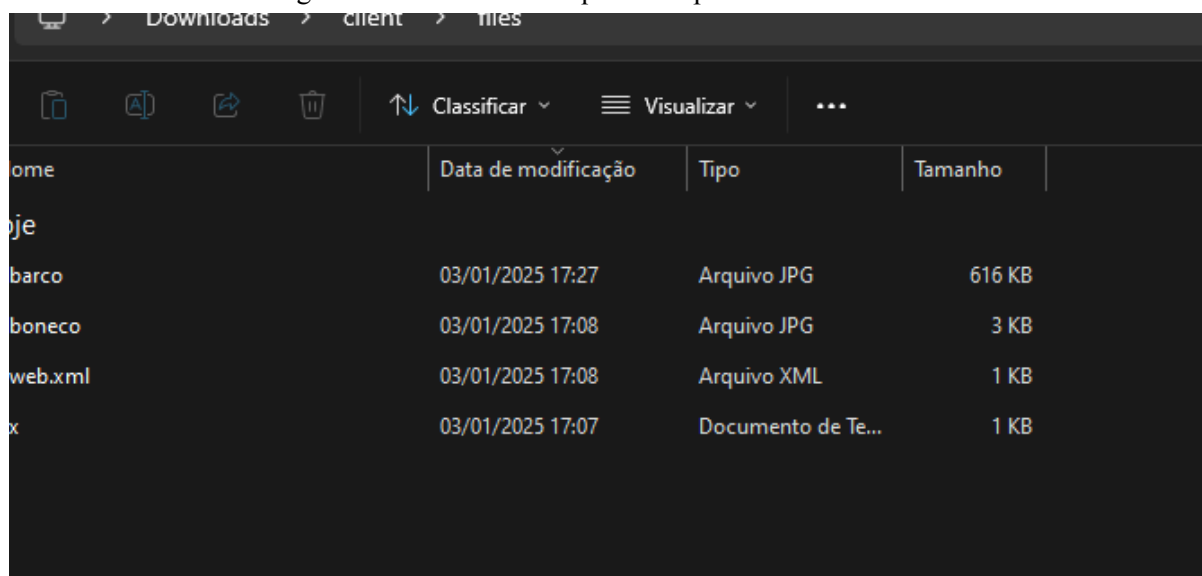
Figura 6: Tamanho original dos arquivos na pasta files no servidor.



Nome	Data de modificação	Tipo	Tamanho
Anteriormente nesta semana			
x	30/12/2024 16:58	Documento de Te...	1 KB
web.xml	30/12/2024 16:58	Arquivo XML	1 KB
boneco	30/12/2024 16:57	Arquivo JPG	3 KB
barco	30/12/2024 16:57	Arquivo JPG	7.332 KB

Fonte: Autoria própria, 2025.

Figura 7: Tamanho dos arquivos na pasta files do cliente.

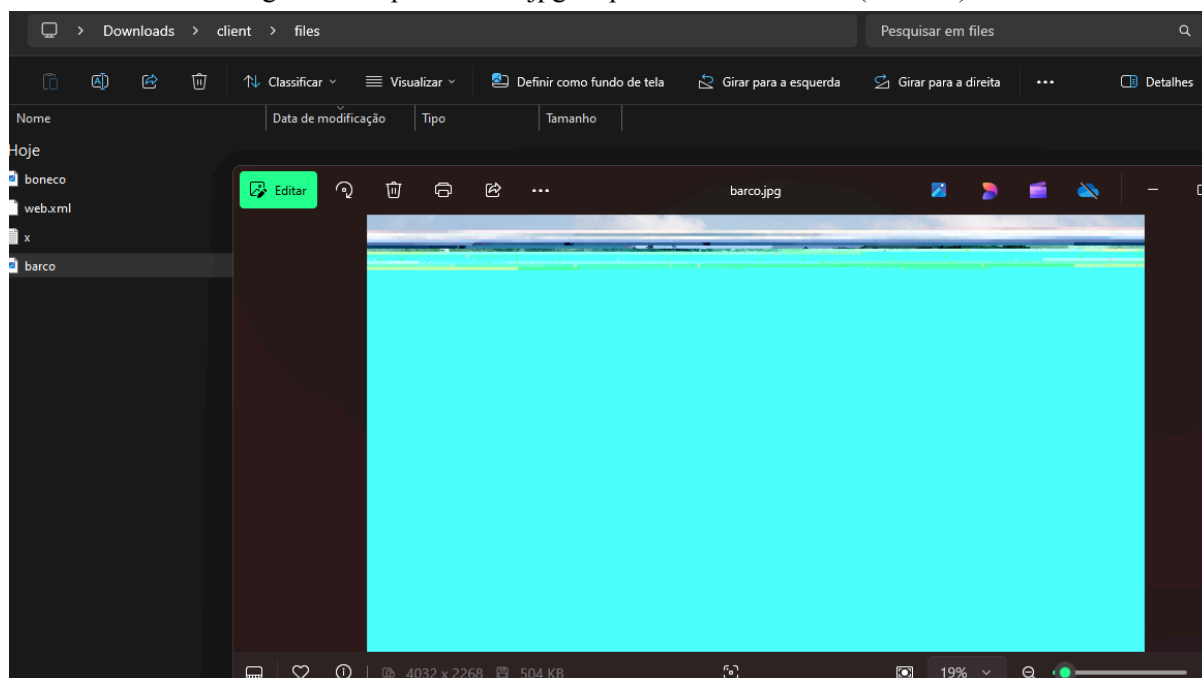


Nome	Data de modificação	Tipo	Tamanho
barco	03/01/2025 17:27	Arquivo JPG	616 KB
boneco	03/01/2025 17:08	Arquivo JPG	3 KB
web.xml	03/01/2025 17:08	Arquivo XML	1 KB
x	03/01/2025 17:07	Documento de Te...	1 KB

Fonte: Autoria própria, 2025.

No teste com blocos de 1024 bytes, repare que o arquivo ‘barco.jpg’ na Figura 7 é o único que não teve seus dados enviados completamente ao cliente, diferente dos demais. Ao abrir o arquivo ‘barco.jpg’ na pasta ‘files’ do cliente, percebe-se que ele está incompleto como mostrado na Figura 8, diferentemente da imagem completa representada na Figura 9.

Figura 8: Arquivo barco.jpg na pasta files do cliente (616 KB).



Fonte: Autoria própria, 2025.

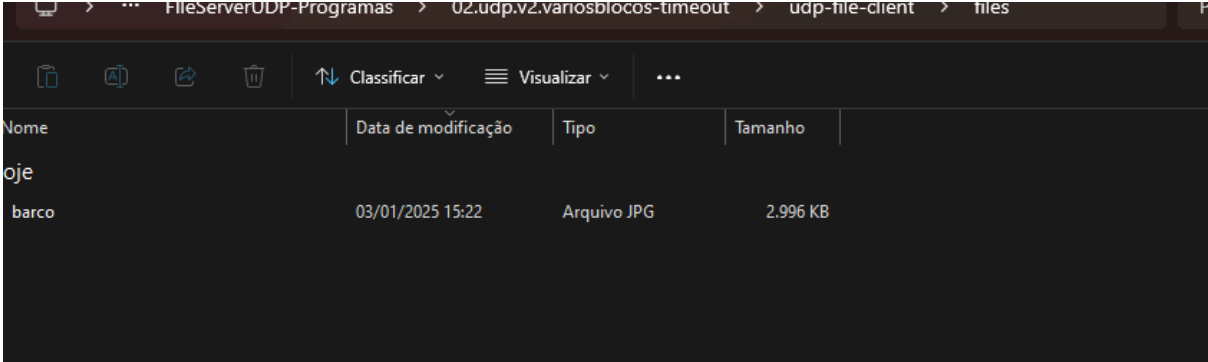
Fonte: Autoria própria, 2025.

[illegible]

Fonte: Autoria própria, 2025.



Figura 12: Tamanho do arquivo barco.jpg com blocos de 2048 bytes.

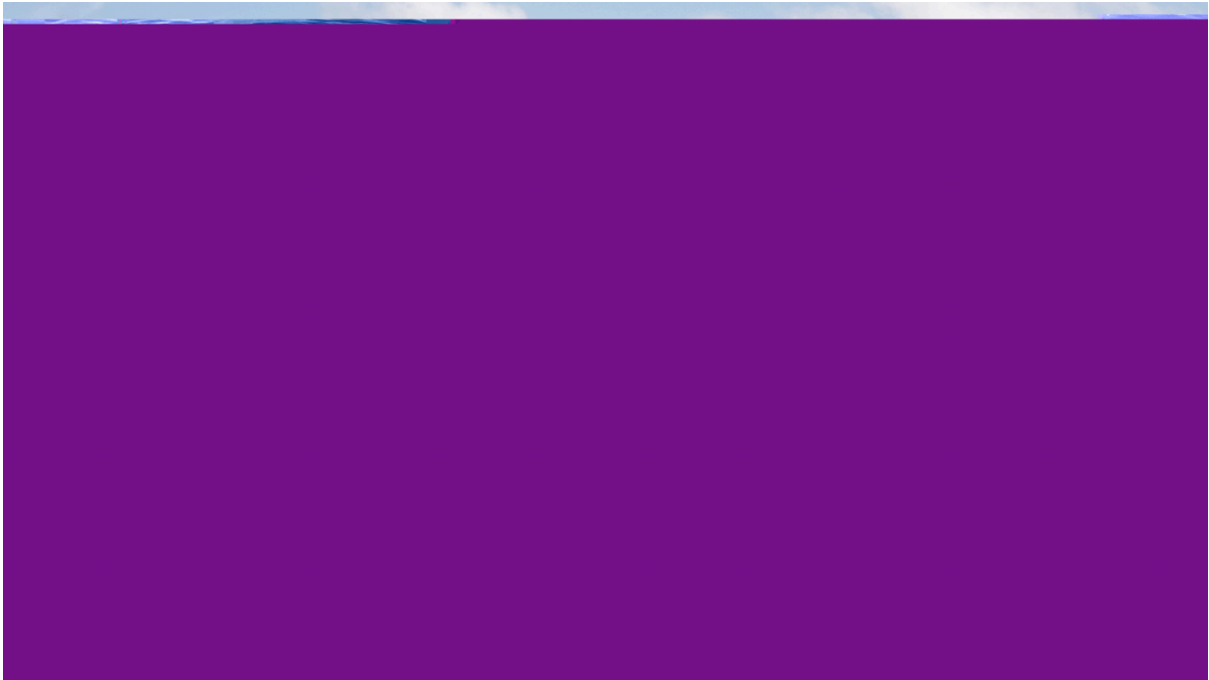


The image shows a file explorer window with a dark theme. The address bar at the top displays the path: FileServerUDP-Programas > 02.udp.v2.variosbloco-timeout > udp-file-client > files. Below the address bar is a toolbar with icons for file operations and a menu. The main area contains a table with the following columns: Nome, Data de modificação, Tipo, and Tamanho. A single file named 'barco' is listed with a modification date of 03/01/2025 15:22, a type of 'Arquivo JPG', and a size of '2.996 KB'.

Nome	Data de modificação	Tipo	Tamanho
barco	03/01/2025 15:22	Arquivo JPG	2.996 KB

Fonte: Autoria própria, 2025.

Figura 13: Arquivo barco.jpg na pasta files do cliente 2.996 KB



Fonte: Autoria própria, 2025.

### III- Conclusão:

Por fim, chegamos a conclusão de que em ambos os casos terão algum tipo de problema, seja por causa do tempo, ou por causa do tamanho dos blocos de bytes, a primeira solução que todos pensariam seria a de aumentar ambos, mas ainda assim continuaria tendo problemas, porque ou teria que esperar bastante tempo mesmo enviando um arquivo pequeno, ou por conta de existir arquivos ainda maiores, tendo que modificar o programa toda vez.