

## Implementação prática em node.js.

1) Criar uma pasta (local de desenvolvimento das aplicações em Node.js), implementar todos exemplos da aula PL.

2) Utilizando o exemplo de servidor da semana passada crie uma aplicação que permita um cliente (browser) adicionar um texto num ficheiro (mensagens.txt). A texto a ser adicionado ao ficheiro faz parte do URL, é o último segmento. Ver fig.1

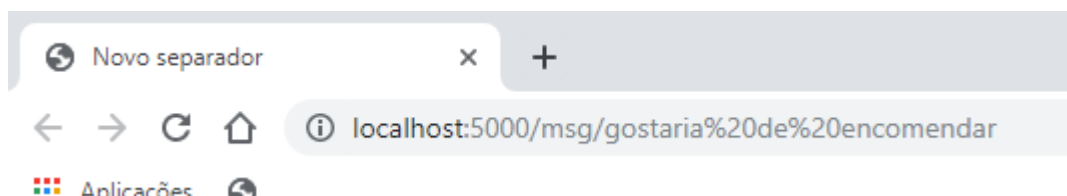


Figura 1. Mensagem a gravar 'gostaria de encomendar'

[Writing files in Node.js](#)

[Acessos concorrentes a ficheiros](#)

3) Ver com detalhe o conteúdo headers do objeto **req** (requisição/solicitação, resposta) do exemplo da PL da semana passada

[https://www.w3schools.com/nodejs/nodejs\\_http.asp](https://www.w3schools.com/nodejs/nodejs_http.asp)

[https://www.w3schools.com/nodejs/obj\\_http\\_incomingmessage.asp](https://www.w3schools.com/nodejs/obj_http_incomingmessage.asp)

4) Criem respostas (em função do tipo de [https://developer.mozilla.org/pt-PT/docs/Web/HTTP/Headers#Content\\_negotiation](https://developer.mozilla.org/pt-PT/docs/Web/HTTP/Headers#Content_negotiation)); <https://developer.mozilla.org/pt-PT/docs/Web/HTTP/Headers/Accept> ) para as seguintes solicitações:

- 1) <http://localhost:5000/estudantes> (Accept: application/xml)
- 2) <http://localhost:5000/professor> (Accept: application/json)

No caso de a linha de solicitação não possuir a propriedade definida nas alíneas 1 e 2 o servidor deve enviar uma mensagem de erro 406 (<https://developer.mozilla.org/pt-PT/docs/Web/HTTP/Status/406>).

5) Em relação ao servidor HTTP (apache, IIS) quais são as vantagens?  
Ver a TP2

Penha, 23 de setembro de 2024