

Ejercicios complementarios relacionados con el laboratorio S04

Crea un proyecto nuevo e introduce en el mismo lo realizado en el laboratorio anterior, respetando la división en paquetes. Trabaja sobre este nuevo proyecto. desarrolla los métodos que se indican, pero intenta reutilizar al máximo el código o, dicho de otra manera, procura que no haya código repetido (y asegúrate que todo lo hecho en S4 sigue funcionando). No prestéis atención a los saltos de línea visualizados en los ejercicios f y g.

1. En la clase **MailServer** implementa:
 - a. **inboxSize**: Dado el nombre de un usuario del sistema, obtiene cuántos mensajes hay para dicho usuario.
 - b. **removeEmailInboxPos**: Dado el nombre de un usuario del sistema y un número n, borra de la lista de emails el email correspondiente al n-ésimo mensaje del Inbox del usuario. Si no existe escribirá sólo un mensaje de error.
 - c. **obtainRemoveEmailInboxPos**: Dado el nombre de un usuario del sistema y un número n, obtiene y borra de la lista de emails el email correspondiente al n-ésimo del Inbox del usuario. Si no existe escribirá sólo un mensaje de error.
2. Realiza las modificaciones que estimes oportunas en la clase **MailClient** para poder implementar correctamente los siguientes métodos:
 - d. **removeEmailFromInboxPos**: Dada una posición n, elimina de la lista de emails del servidor el n-esimo mensaje del Inbox del usuario.
 - e. **obtainRemoveEmailFromInboxPos**: Dada una posición n, elimina de la lista de emails del servidor el n-esimo mensaje del Inbox del usuario.
 - f. **replyEmail**: Dado un mensaje y un texto, responde al mensaje con un mensaje nuevo. El mensaje debe tener el sender y receiver apropiados, en el asunto se debe concatenar por delante el String "RP: " y en el texto deberá aparecer el texto dado seguido de la información del sender, receiver y asunto original. Ver ejemplo:

Mensaje original

Sender: maite.urretavizcaya@ehu.eus
Receiver: julian.gutierrez@ehu.eus
Subject: lab S3 y S4
Text: Ya está en eGela el enunciado del laboratorio de la semana 3

Mensaje respondido

Sender: julian.gutierrez@ehu.eus
Receiver: maite.urretavizcaya@ehu.eus
Subject: **RP**: lab S3 y S4
Text: Lo reviso y te indico. Un saludo.

Sender: maite.urretavizcaya@ehu.eus
Receiver: julian.gutierrez@ehu.eus
Subject: lab S3 y S4
Ya está en eGela el enunciado del

- g. **resendEmail**: Dados un mensaje y un usuario receptor, reenvía el mensaje (algo modificado) a dicho usuario. El mensaje modificado debe tener en el asunto concatenado por delante el String "RS: " y en el texto deberá aparecer el texto dado seguido de la información del sender, receiver y asunto original. Por ejemplo, para el mensaje original anterior si lo quiero reenviar a juanmiguel.lopez@ehu.eus la información del mensaje podría ser la que se muestra.

Mensaje reenviado

Sender: maite.urretavizcaya@ehu.eus
Receiver: juanmiguel.lopez@ehu.eus
Subject: **RS**: lab S3 y S4
Text: De acuerdo con el enunciado

Sender: maite.urretavizcaya@ehu.eus
Receiver: julian.gutierrez@ehu.eus
Subject: lab S3 y S4
Ya está en eGela el enunciado del laboratorio

3. Prueba en una clase **Lab04Complementarios** el correcto funcionamiento de todos los métodos implementados.
4. Más cuestiones de diseño: ¿Cómo puedes saber quiénes son clientes del servidor y te asegures que se envían mensajes sólo entre Clientes? Cada vez que se lee un mensaje del inbox este desaparece del servidor (y así debe seguir siendo), pero ¿cómo lo podría recuperar el cliente? ¿quizás almacenándolo él mismo? ¿cómo cambiaría tus implementaciones para incluir estos nuevos requisitos?