

PRÁCTICA 6 – CHAT MULTIUSUARIOS

Asignatura: Diseño Arquitectónico y Patrones

Alumno: Mauricio José Orta Rodríguez.

INTRODUCCIÓN

El enunciado para esta práctica fue el siguiente:

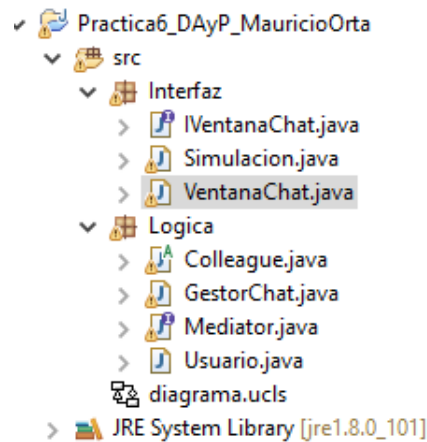
«Desarrolle un programa en java que implemente un chat que permita la comunicación en tiempo real entre varios usuarios. Los usuarios podrán escribir mensajes en su teclado, y el texto aparecerá automáticamente y al instante en las ventanas de todos los participantes.

El programa deberá estar bien estructurado, atendiendo a todos los principios de diseño aprendidos en clase y presentar entorno gráfico».

A continuación, se muestra un resumen de la estructura del código construido y su diagrama de clases. Asimismo, se detallan los patrones de diseños aplicados durante su elaboración.

ESTRUCTURA DE CÓDIGO

El código se divide en dos paquetes: Logica e Interfaz.



Paquete Logica

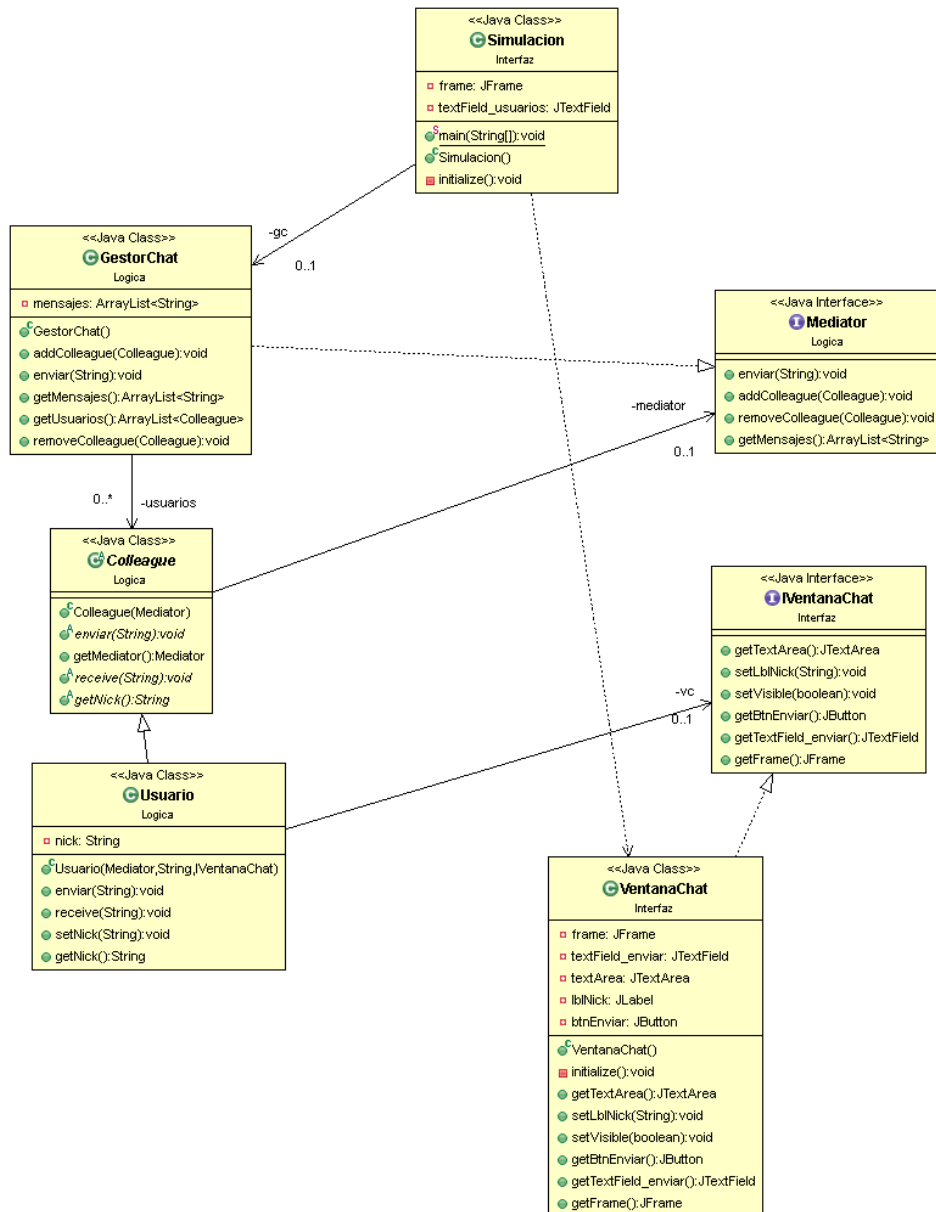
Contiene una interfaz Mediator, que define un método de envío de mensajes a objetos colegas, y una clase abstracta Colleague, que representa a los objetos colega y declara sus métodos de envío de mensajes a un Mediator y la recepción de los que este envía a él.

La clase Usuario implementa la interfaz Colleague, al ser el conjunto de objetos colegas en este sistema, puesto que todos ellos deben interactuar entre sí y estar al tanto de lo que han enviado. GestorChat implementa la interfaz Mediator y gestiona el intercambio de mensajes de los Usuarios, garantizando que todos los mensajes que le son enviados por cada Usuario, lleguen y se muestren en la ventana de chat (el cual es un atributo de dicha clase) de ellos y los demás como se espera.

Paquete Interfaz

Se compone de una clase Simulacion, la cual crea una interfaz en donde se determina el número n de usuarios del chat simulado y los nombres o “nicks” de cada uno. Al terminar de fijar esos parámetros, se crean las n ventanas de cada uno representadas como objetos de la clase VentanaChat. Cada usuario específico se comunica y asocia a una ventana de chat respectiva por medio de los métodos de la interfaz IVentanaChat.

Diagrama de clases



PATRONES DE DISEÑO UTILIZADOS

- **Patrón Mediator:** define un objeto mediador que encapsula y centraliza las interacciones entre un conjunto de objetos colegas.

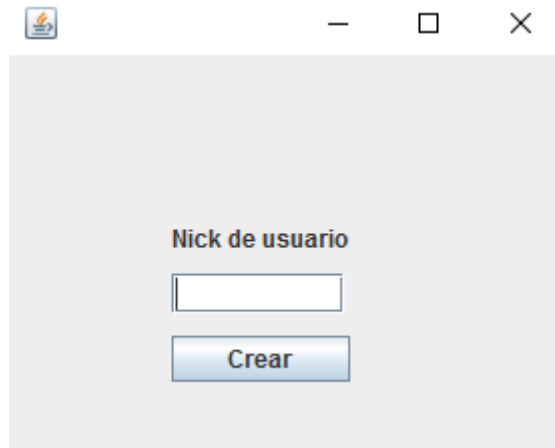
La estructura de clases del paquete Logica sigue este patrón, con el GestorChat actuando como el mediador entre los distintos objetos Usuario colegas que se intercambian mensajes entre sí. Todos los mensajes enviados por los usuarios son recibidos por el gestor y este se encarga de dirigirlos a cada una de las ventanas de chat asociadas a cada uno de ellos.

DISEÑO DE INTERFAZ GRÁFICA

La interfaz tiene tres partes, cuyo orden de uso y aparición es el siguiente:

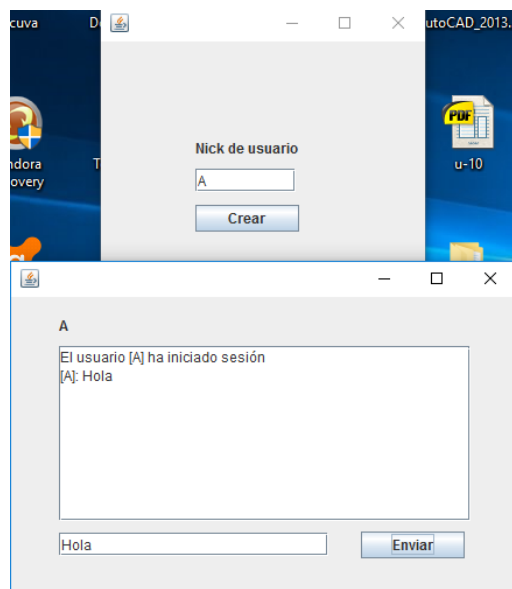
1.) Nick de usuario

Se pide introducir el nick de un usuario del chat antes de crear su ventana.



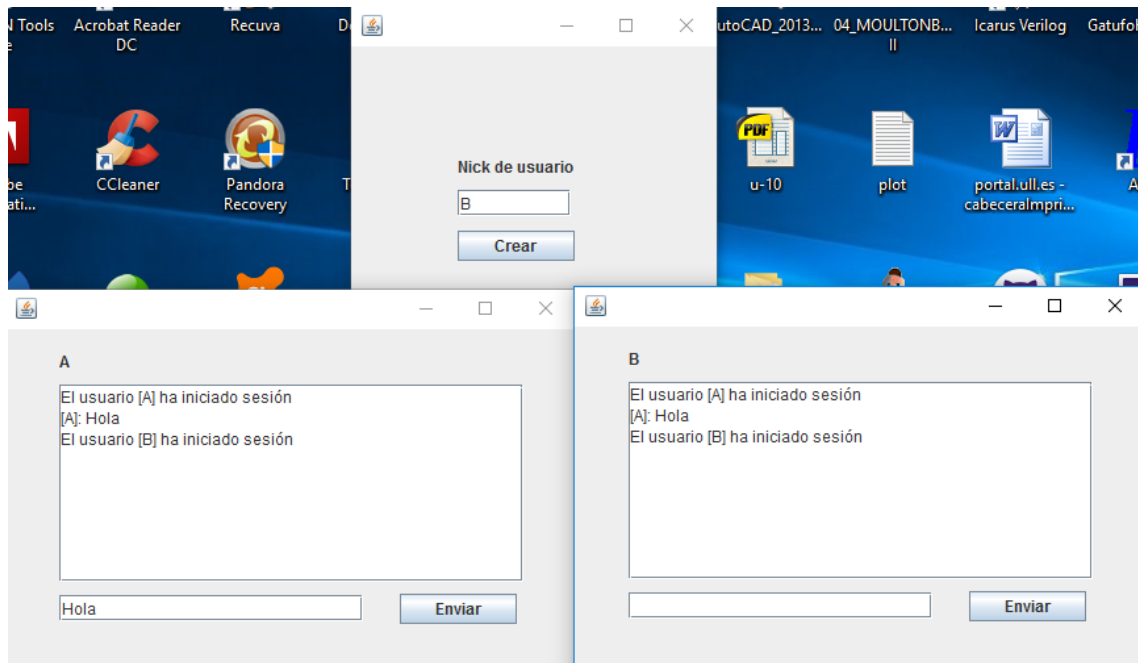
2.) Ventanas de chat

Después de asignar a cada usuario su Nick, damos click a “Abrir” y su ventana del chat será creada.



Al escribir en el campo de Input de cada ventana y presionar “Enviar”, el mensaje del usuario aparecerá en su ventana de chat.

Podemos crear nuevas ventanas para nuevos usuarios como se explica al principio. Estos nuevos usuarios recibirán todos los mensajes que se hayan enviado los usuarios conectados al chat antes que él.



Los mensajes escritos por cada usuario serán visibles en su ventana y en las de los demás conectados al chat.

